Homecomputer

5 1. Jahrgang

Mai '83 **5,50 DM** 45 öS 6,00 sfr



In diesem Heft u.a.:

VC-20
Editor
Assembler
HC-Invaders
Car-Crash

Sharp MZ-80K
Eidechse

VC-64
Geisterjäger
Mondlandung
Wurm
Sprite-Generator

ZX-81
Ritter
Missile ZX-Command

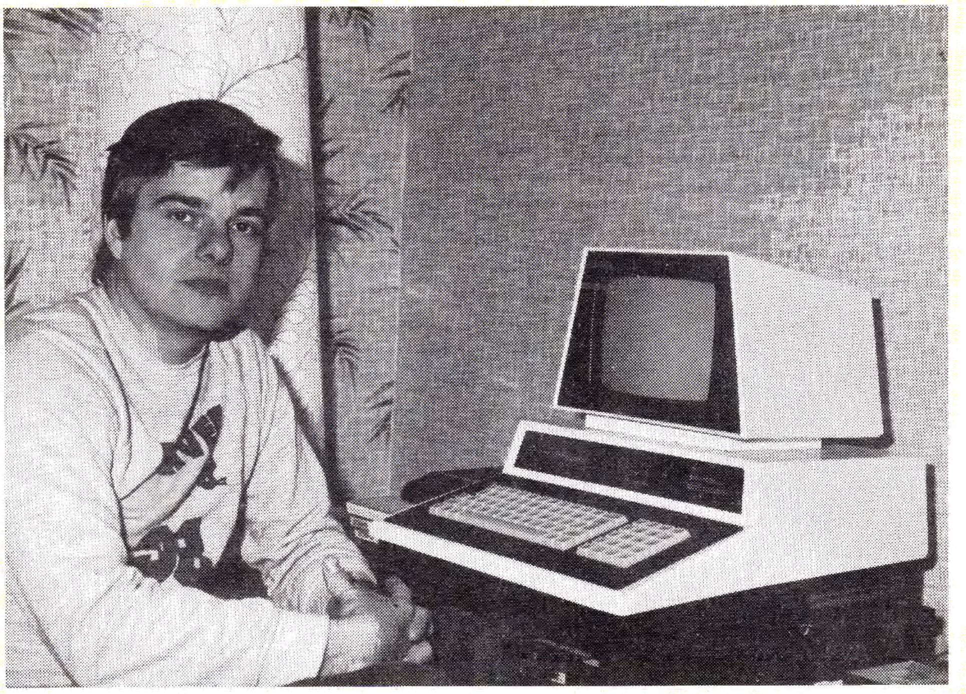
Apple II
Disk-Schutz 2

CBM
Mastermind (Computerhirn)
Mastermind II

Texas-Special
mit vielen Programmen für den TI-99/4A

Jeden Monat mehr als 15 neue Programme + Tips + News aus aller Welt

Große Werbeaktion! 250 Computer zu gewinnen!



Liebe Leserin, lieber Leser,

wenn Sie diese Zeilen lesen, ist die Hannover-Messe 83 schon beendet.
Jetzt im Moment, in dem ich diese Zeilen schreibe, steht sie wenige Stunden bevor. 8 Tage lang werden wir uns bemühen, das Interessanteste für Sie, die Leser von Homecomputer, herauszupicken, werden wir versuchen, detaillierte Informationen über alles Neue und Berichtenswerte zusammenzufassen.
Im Juni-Heft werden wir dann darüber berichten. Darum wird die Neuheitenrubrik etwas vergrößert werden. Allerdings würde ein allumfassender Bericht den Rahmen von Homecomputer bei weitem sprengen und weniger veröffentlichte Software bedeuten.
Dies möchten wir nicht. Auch in Heft 6/83 sollen, wie bei uns üblich, mehr als 15 Programme enthalten sein.
Aus diesem Grunde bringen wir ein „Sonderheft Hannover-Messe" heraus, damit Sie sich – auch falls Sie nicht die Gelegenheit hatten länger oder überhaupt diese wichtigste aller Messen zu besuchen – über alles ausführlich informieren können.
Microcomputer, Drucker, Datenspeichergeräte fürs Heim; darüber werden wir in Wort und Bild berichten und hoffen, Ihnen damit eine interessante Ergänzung zum gewohnten Homecomputer mit dem Programmservice zu bieten.

Herzlichst
Ihr Ralph Roeske

Ralph Roeske

Herausgeber, Chefredakteur

Impressum

Homecomputer erscheint monatlich im:
Roeske Verlag, Eschwege

Herausgeber:
Ralph Roeske

Redaktion:
Ralph Roeske (R), Chefredakteur (verantwortlich)
Cornelius Geppert (Ge)
Bärbel Voigt

Herstellung:
Roeske Verlag, Eschwege

Satz und Reproduktion:
Imtal Composing GmbH
3436 Hessisch Lichtenau

Druck:
Vogt GmbH
Druckerei und Verlag
3436 Hessisch Lichtenau

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel), sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Str. 7
6200 Wiesbaden
Tel: 06121-2660

Anzeigenleitung:
Monika Roeske

Erscheinungsweise:
Erstverkaufstag von Homecomputer ist jeweils der letzte Montag vor dem Erscheinungsmonat

Anschrift:
Homecomputer
Gartenstraße 47
3440 Eschwege
Tel: 05651-5993

Bezugspreise:
Einzelheft: 5,50 DM
Abonnement: Inland 55,- DM pro Jahr (12 Ausgaben)
Ausland: Europa 80,- DM
USA 110,- DM

Anzeigenpreise:
Es gilt die Anzeigenpreisliste Nr. 1 vom 1.Januar 1983
1/1 Seite S/W 3000,- DM
Farbzuschlag je Farbe (Europaskala) 500,- DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne Entgegen. Honorare nach Vereinbarung. Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Rücksendung erfolgt nur bei angeforderten Beiträgen, ansonsten nur gegen Erstattung der Unkosten.
Zusendungen von Software zur Veröffentlichung sollte bitte folgendes enthalten:
Kopierfähige Kassette oder Diskette mit dem Programm, von Drucker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreimaschinen-Listings), eventuell Bildschirmfotos von einem Probelauf.

Inhalt

Homecomputer bringt in diesem Monat:

Heft 3 noch lieferbar

Ich bin angenehm beeindruckt von dem Inhalt Ihrer Zeitschrift Homecomputer. Leider habe ich das erste Heft nicht mehr am Kiosk bekommen. Bitte senden Sie mir das Heft 3/83 und die VC20-Kassette Heft 4.

P. Hirsch — Köln

Es ist ab Mai noch eine begrenzte Zahl von Heft 3 zu haben. Bitte bestellen Sie über den Verlag.

zu Slalom Heft 3/83

Zu Heft 3 habe ich folgende Frage:
Wie steuert man im Programm Slalom den Fahrer?

Johann Breitenfellner — Linz

Die Tasten Z und / steuern den Alpinisten. Das haben wir vergessen abzudrucken.

ZX-Drucke

Mit großem Interesse habe ich Ihre Zeitschrift "Homecomputer" gelesen.
Beim Labyrinth-Programm für den ZX81 treten bei mir jedoch Schwierigkeiten auf. Das Unterprogramm in Maschinensprache, welches in einem Stringausdruck untergebracht ist, ist leider kaum lesbar.
Die Zeichen 8 und B sind teilweise nicht zu unterscheiden. Aus diesem Grund läuft das Unterprogramm bei mir nicht.
Mein Tip für zukünftige Ausgaben: Geben Sie das Programm in Maschinensprache in Dezimal an (wie bei U-Boot-Jagd), da dort eine Verwechselung zwischen Bund 8 nicht auftreten kann.

Thomas Kaus — Burladingen 1

Negativ...

Positiv finde ich ...
Negativ fällt es daher auf, daß Eure Zeitschrift eben gar nicht niedriger im Preis rangiert. Sie ist sogar, vergleicht man Preis und Umfang, so ziemlich die teuerste deutsche Zeitschrift ihrer Art! (z.B. kostet Computer persönlich nur 4,– DM!)
Anzeigen in denen sich der Leser über den deutschen Geräte- und Softwaremarkt informieren kann, suchte ich in Homecomputer so gut wie vergeblich. Das ist nicht nur eine inhaltliche Lücke (ich kaufe eine Fachzeitschrift gerade auch wegen der Anzeigen!) - es heißt auch, daß der Käufer die Herstellungskosten alleine zu decken hat!

Franz Nagel — Darmstadt

**Besitzen Sie einen Homecomputer?
Sicher nicht, sonst hätten Sie das Preis/Leistungsverhältnis (von dem in der Computerbranche oft die Rede ist) schon erkannt. Für einen VC-20-Anwender z.B. sind allein die Programme der ersten drei Ausgaben von Homecomputer einige Jahresabos wert – und das bei nur 16.50 DM!**

Nicht vollständig?

Wäre es nicht möglich, in zukünftigen Ausgaben alle (auch öde Tippereien wie Apple-Steno) Programme vollständig abzudrucken, da 1 Disk/Monat die Taschengeldgrenze manches Kleincomputeraners haushoch übersteigt.

Wolfgang Schmidt — München

Teilen Sie uns bitte mit, welche Seite in Ihrem Heft gefehlt hat. Oder haben Sie etwa nur den Absatz zum Leser-Service auf Seite 33 mißverstanden?

Umsonst geärgert – Handbuch lesen.

Die Märzausgabe von Homecomputer hat mich 505,50 DM gekostet! 5,50 für das Heft und für 500,00 DM habe ich mich geärgert. Ich habe nämlich das Robotprogramm (fehlerfrei!) eingetippt und dazu brauche ich ja anscheinend keine 16K-Erweiterung. Trotzdem läuft das Programm nicht. Nachdem die Leertaste gedrückt wurde ist es aus und vorbei, Errors, so weit das Auge reicht. Da ich fehlerfrei getippt habe (das habe ich bis zur "Vergasung" nachgeprüft), glaube ich nicht, daß der Fehler bei mir liegt.

Thomas Schiefer — Eggenfelden

**Und doch ist es so. Weil die meisten VC-20-Besitzer eine Speichererweiterung um mindestens 8 K besitzen, schreiben wir Programme fast immer für 8- oder 16-K-Versionen.
Daß dann die Adressen für Video- und Farbmatrix angepaßt werden müssen, wenn man weniger als 11.5 K freien Speicher hat, ist aus dem Handbuch zu ersehen.
Der VC-20 ist aber glückleicherweise der einzige Microcomputer, der uns auf diese Art und Weise angeschmiert hat, denn bei anderen Computern stellt sich das Problem nicht.**

MID$ mit Dreien?

Bei Ihrem Kalender-Programm für den PET bin ich mit Ihrer Zeile 135 nicht ganz einverstanden. MID$ hat doch 3 Argumente. Ich kann leider aus Zeitgründen nicht das ganze Programm durchgehen und möchte Sie bitten den Fehler zu korrigieren.

Stefan Munch — Erlensee

Gibt es etwas zu korrigieren? Wer das Computerhobby ernsthaft betreibt, muß sich hin und wieder ein wenig Zeit nehmen, zu lernen, was noch nicht gelernt oder verstanden wurde. Probieren Sie den MID$ mit 2 Argumenten aus. Sie werden überrascht sein.

Ferien im Computer-Camp

Super Freizeitspaß mit Computer-learning für junge Leute!

Im Luftkurort "Haltern am See" findet am 310 ha großen Trinkwassersee der Stevertalsperre eines der ersten europäischen Computercamps statt. Junge Leute werden hier spielerisch mit dem Mikrocomputer und der Programmiersprache BASIC vertraut gemacht. Durch intensive Beschäftigung mit dem Mikrocomputer COMMODORE (für je zwei Teilnehmer steht eine Commodore-Anlage zur Verfügung) wird die Angstschwelle schnell abgebaut und der Umgang mit der EDV-TECHNIK zur Selbstverständlichkeit.
Vorgesehen sind: 50 Unterrichtsstunden Computerlearning unter Anleitung von Fachdozenten, umfangreiches Unterrichtsmaterial, Einsatz von Video, zusätzlicher theoretischer Unterricht, Einführung in die EDV und Mikrocomputer, sowie deren Möglichkeiten und Grenzen. Neben dem Computerlearning steht der Kontakt der Teens und Twens mit Sport, Spiel und Freizeitspaß im Vordergrung. Das Freizeit-Aktionsprogramm bietet (teilweise gegen geringe Unkostenbeteiligung) Fußball, Schwimmen, Reiten, Fahrradverleih u. a.. Für das Rundumprogramm sind weiterhin vorgesehen: Planwagenfahrten, Nachtwanderungen, Lagerfeuer und heiße Disconächte. Die Unterbringung, die gemeinsamen Mahlzeiten, die Programmierung der Mikrocomputer COMMODORE findet in der "Jugendherberge Haltener-Stausee" Overrath 39, 4358 Haltern-Stausee, statt.

111 Cassetten im Programm

BUG-BYTE SOFTWARE
AUTHORISED DEALER

Software-Anbieter für die populären Homecomputer sind zur Zeit noch nicht sehr zahlreich gesät. Einer der größten Anbieter für ZX 81, Spectrum, VC 20, Acorn, BBC und Dragon ist der Softwareversand Augsburg, der jetzt auch den Spectrum-Computer im Programm hat. Produkte der Firmen Artic, Bug Byte, Melbourne House, Publishers Personal Software A+F und Impact werden zum Teil exclusiv vertrieben.

Alles mit ausführlicher (allerdings englischer) Beschreibung und original verpackt. Ein Katalog mit dem vorhandenen Programmspektrum ist gegen Gebühr erhältlich.

Softwareversand Thomas Wagner, Augsburg

Microdrive bald lieferbar.

Noch immer wartet die Homecomputerwelt auf die Fertigstellung des vor 1½ Jahren angekündigten Microdrives.

Laut Sinclair stehen lediglich noch die speziell für die Firma hergestellten Halbleiter-Chips aus, die zur Zeit noch eine Testphase durchlaufen. Das Design steht ebenfalls und in Kürze sollen für £40 per Stück die ersten 100.000 Geräte zunächst an die ersten Spectrum-Kunden geliefert und dann über den Handel auch der Rest der Interessenten bedient werden.
Somit wird dann auch ein Gerücht sein Ende finden, nämlich, ob es sich um Microfloppies oder um "Microbänder", ähnlich den sogenannten "Stringy Floppies", handelt. Laut unseren letzten vorliegenden Informationen handelt es sich um ein Endlos-Band, das sich mit hoher Geschwindigkeit innerhalb 7 Sekunden einmal am Übertragungskopf vorbeibewegt. So wird eine durchschnittliche Zugriffszeit von 3,5 Sekunden erreicht.

TRS-80/VG-Club in Wuppertal

Zu unserem Aufruf anläßlich der Clubecke im letzten Heft sandte uns Herr Gerd Breidenbruch aus Wuppertal folgende Mitteilung.

Der TRS-80/Mod.1 hat meines Erachtens einen sehr großen Marktanteil und ist immerhin der 'Homecomputer mit dem größten Software-Angebot'
Vielleicht interessiert es Sie in diesem Zusammenhang, daß es in Wuppertal einen TRS-80/Video-Genie Club gibt (Anzahl der Mitglieder zur Zeit ca. 40, eigene Clubzeitschrift, Clubbeitrag keiner).
Die Mitglieder des TRS-80/VG Clubs treffen sich monatlich einmal in der Universität Wuppertal Raum 113.34. Jeder Computer-Fan ist dort herzlich willkommen.
Da die Treffen noch in unregelmäßigen Abständen stattfinden, sollten sich Interessenten erst bei dem Leiter der Gruppe, Herrn Horst Brodowski, Gördelerstr. 7, 5657 Haan/Rhl. Tel.: 02129/4891 erkundigen.

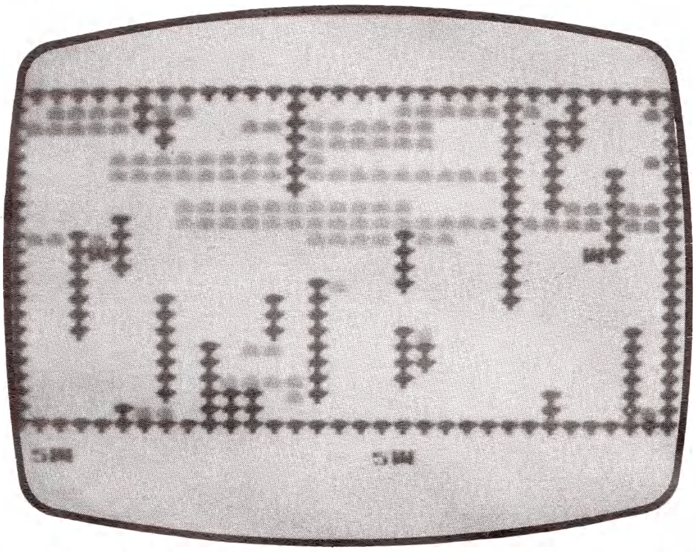
Wir bedanken uns für die freundliche Unterstützung bei der Titelgestaltung beim
Round up Western Shop
Frankfurter Straße 88 · 3500 Kassel

Panzerkrieg

TI-99/4A + Fernbedienungen

Der Sinn dieses Zwei-Personen-Spieles besteht darin, die gegnerischen Panzer abzuschießen und dabei möglichst wenig Flur- und Sachschaden anzurichten. Am Anfang stehen sich die Feinde gegenüber, getrennt durch Baum- und Häuserreihen. Die Häuser können zwar abgeschossen werden, bringen aber 300 Minuspunkte ein. Will man ein Haus umfahren, wird der Panzer zerstört und der Gegner bekommt 1000 Pluspunkte. Ebenso bei Abschuß. Die Bäume können weder abgeschossen noch umgefahren werden, "wehren" sich jedoch mit Minuspunkten.
Anm.: Die Zeilen 740-930 verbinden, daß ein Panzer am Anfang von Dämmen umgeben und bewegungsunfähig ist. Sollten die Gegner jedoch durch eine durchgehende Baumreihe getrennt sein, muß das Spiel mit "CLEAR" abgebrochen und mit "RUN" neu gestartet werden.

Jürgen Hoppstädter



```
>LIST -210
 100 RANDOMIZE
 110 REM  ZEICHEN DEFINIEREN
 120 CALL CHAR(128,"DBDBDBFFF
FFFFFC3")
 130 CALL CHAR(129,"FFFF787F7
F78FFFF")
 140 CALL CHAR(130,"C3FFFFFFF
FDBDBDB")
 150 CALL CHAR(131,"FFFF1EFEF
E1EFFFF")
 160 CALL CHAR(132,"000000181
8000000")
 170 CALL CHAR(133,"00003C242
43C")
 180 CALL CHAR(134,"007E42424
2427E")
 190 CALL CHAR(135,"FF8181818
18181FF")
 200 CALL CHAR(136,"DBDBDBFFF
FFFFFC3")
 210 CALL CHAR(137,"FFFF787F7
F78FFFF")
>
```

```
 220 CALL CHAR(138,"C3FFFFFFF
FDBDBDB")
 230 CALL CHAR(139,"FFFF1EFEF
E1EFFFF")
 240 CALL CHAR(140,"000000181
8000000")
 250 CALL CHAR(141,"00003C242
43C")
 260 CALL CHAR(142,"007E42424
2427E")
 270 CALL CHAR(143,"FF8181818
18181FF")
 280 CALL CHAR(144,"3C7FFF7E7
C181818")
 290 CALL CHAR(152,"00387CFEF
EFEFEFE")
 300 CALL CHAR(153,"00003C242
43C")
 310 CALL CHAR(154,"007E42424
2427E")
 320 CALL CHAR(155,"FF8181818
18181FF")
 330 REM  FARBEN DEFINIEREN
>
```

```
>LIST 340-550
 340 CALL SCREEN(15)
 350 CALL COLOR(13,5,1)
 360 CALL COLOR(14,14,1)
 370 CALL COLOR(15,13,1)
 380 CALL COLOR(16,10,1)
 390 REM  HAEUSER BAUEN
 400 CALL CLEAR
 410 P1=0
 420 P2=0
 430 FOR I=1 TO 30
 440 X=23*RND+1
 450 Y=31*RND+1
 460 Z=9*RND+1
 470 CALL HCHAR(X,Y,152,Z)
 480 NEXT I
 490 REM  BAEUME PFLANZEN
 500 FOR I=1 TO 30
```

```
 510 X=23*RND+1
 520 Y=31*RND+1
 530 Z=7*RND+1
 540 CALL VCHAR(X,Y,144,Z)
 550 NEXT I
>
```

```
>LIST 560-750
 560 REM  RAND/PUNKTETABELLE
 570 CALL HCHAR(24,1,32,32)
 580 CALL HCHAR(23,1,32,32)
 590 PANZER1=5
 600 PANZER2=5
 610 GOSUB 4560
 620 GOSUB 4590
 630 CALL HCHAR(1,1,144,32)
 640 CALL HCHAR(22,1,144,32)
 650 CALL VCHAR(1,1,144,22)
 660 CALL VCHAR(1,32,144,22)
 670 CALL HCHAR(24,3,128)
 680 CALL HCHAR(24,19,136)
 690 REM   STARTPOSITION
 700 XP1=11
 710 XP2=11
 720 YP1=5
 730 YP2=28
 740 CALL GCHAR(XP1-1,YP1,ZP1
)
 750 IF ZP1<>144 THEN 840
>
```

```
 760 CALL GCHAR(XP1,YP1+1,ZP1
)
 770 IF ZP1<>144 THEN 840
 780 CALL GCHAR(XP1+1,YP1,ZP1
)
 790 IF ZP1<>144 THEN 840
 800 CALL GCHAR(XP1,YP1-1,ZP1
)
 810 IF ZP1<>144 THEN 840
 820 YP1=YP1+1
 830 GOTO 740
 840 CALL GCHAR(XP2-1,YP2,ZP2
)
 850 IF ZP2<>144 THEN 940
 860 CALL GCHAR(XP2,YP2+1,ZP2
)
 870 IF ZP2<>144 THEN 940
 880 CALL GCHAR(XP2+1,YP2,ZP2
)
 890 IF ZP2<>144 THEN 940
 900 CALL GCHAR(XP2,YP2-1,ZP2
)
 910 IF ZP2<>144 THEN 940
>
```

```
 920 YP2=YP2-1
 930 GOTO 840
 940 CALL HCHAR(XP1,YP1,128)
 950 CALL HCHAR(XP2,YP2,136)
 960 REM    SPIELHAUPTPROGRAM
M
 970 CALL GCHAR(XP1,YP1,ZP1)
 980 CALL JOYST(1,X1,Y1)
 990 IF Y1<>4 THEN 1010
1000 GOSUB 1240
```

```
1010 IF X1<>4 THEN 1030
1020 GOSUB 1380
1030 IF Y1<>-4 THEN 1050
1040 GOSUB 1510
1050 IF X1<>-4 THEN 1070
1060 GOSUB 1640
1070 CALL KEY(2,K2,S2)
1080 IF K2<>18 THEN 1100
1090 GOSUB 3540
1100 CALL KEY(1,K1,S1)
1110 IF K1<>18 THEN 1130
1120 GOSUB 2660
1130 CALL GCHAR(XP2,YP2,ZP2)
>
```

```
1140 CALL JOYST(2,X2,Y2)
1150 IF Y2<>4 THEN 1170
1160 GOSUB 1770
1170 IF X2<>4 THEN 1190
1180 GOSUB 1910
1190 IF Y2<>-4 THEN 1210
1200 GOSUB 2040
1210 IF X2<>-4 THEN 1230
1220 GOSUB 2170
1230 GOTO 970
1240 REM  FORTBEW. PANZER 1
1250 REM  NACH OBEN
1260 IF ZP1=128 THEN 1300
1270 CALL SOUND(-1000,-8,5)
1280 CALL HCHAR(XP1,YP1,128)
1290 GOTO 1370
1300 CALL GCHAR(XP1-1,YP1,H1
)
1310 IF H1=144 THEN 2300
1320 CALL HCHAR(XP1,YP1,32)
1330 IF H1=152 THEN 2340
1340 XP1=XP1-1
>
```

```
>LIST 1350-1550
1350 CALL SOUND(-1000,-8,5)
1360 CALL HCHAR(XP1,YP1,128)
1370 RETURN
1380 REM  NACH RECHTS
1390 IF ZP1=129 THEN 1430
1400 CALL SOUND(-1000,-8,5)
1410 CALL HCHAR(XP1,YP1,129)
1420 GOTO 1500
1430 CALL GCHAR(XP1,YP1+1,H1
)
1440 IF H1=144 THEN 2300
1450 CALL HCHAR(XP1,YP1,32)
1460 IF H1=152 THEN 2340
1470 YP1=YP1+1
1480 CALL SOUND(-1000,-8,5)
1490 CALL HCHAR(XP1,YP1,129)
1500 RETURN
1510 REM  NACH UNTEN
1520 IF ZP1=130 THEN 1560
1530 CALL SOUND(-1000,-8,5)
1540 CALL HCHAR(XP1,YP1,130)
1550 GOTO 1630
>
```

```
1560 CALL GCHAR(XP1+1,YP1,H1
)
1570 IF H1=144 THEN 2300
1580 CALL HCHAR(XP1,YP1,32)
1590 IF H1=152 THEN 2340
1600 XP1=XP1+1
1610 CALL SOUND(-1000,-8,5)
1620 CALL HCHAR(XP1,YP1,130)
1630 RETURN
1640 REM  NACH UNTEN
1650 IF ZP1=131 THEN 1690
1660 CALL SOUND(-1000,-8,5)
1670 CALL HCHAR(XP1,YP1,131)
1680 GOTO 1760
1690 CALL GCHAR(XP1,YP1-1,H1
)
1700 IF H1=144 THEN 2300
1710 CALL HCHAR(XP1,YP1,32)
1720 IF H1=152 THEN 2340
1730 YP1=YP1-1
1740 CALL SOUND(-1000,-8,5)
1750 CALL HCHAR(XP1,YP1,131)
1760 RETURN
>
```

```
1770 REM  FORTBEW. PANZER 2
1780 REM  NACH OBEN
1790 IF ZP2=136 THEN 1830
1800 CALL SOUND(-1000,-4,5)
1810 CALL HCHAR(XP2,YP2,136)
1820 GOTO 1900
1830 CALL GCHAR(XP2-1,YP2,H2
1840 IF H2=144 THEN 2480
1850 CALL HCHAR(XP2,YP2,32)
1860 IF H2=152 THEN 2520
1870 XP2=XP2-1
1880 CALL SOUND(-1000,-4,5)
1890 CALL HCHAR(XP2,YP2,136)
```

```
1900 RETURN
1910 REM  NACH RECHTS
1920 IF ZP2=137 THEN 1960
1930 CALL SOUND(-1000,-4,5)
1940 CALL HCHAR(XP2,YP2,137)
1950 GOTO 2030
1960 CALL GCHAR(XP2,YP2+1,H2
)
1970 IF H2=144 THEN 2480
>
```

```
1980 CALL HCHAR(XP2,YP2,32)
1990 IF H2=152 THEN 2520
2000 YP2=YP2+1
2010 CALL SOUND(-1000,-4,5)
2020 CALL HCHAR(XP2,YP2,137)
2030 RETURN
2040 REM  NACH UNTEN
2050 IF ZP2=138 THEN 2090
2060 CALL SOUND(-1000,-4,5)
2070 CALL HCHAR(XP2,YP2,138)
2080 GOTO 2160
2090 CALL GCHAR(XP2+1,YP2,H2
)
2100 IF H2=144 THEN 2480
2110 CALL HCHAR(XP2,YP2,32)
2120 IF H2=152 THEN 2520
2130 XP2=XP2+1
2140 CALL SOUND(-1000,-4,5)
2150 CALL HCHAR(XP2,YP2,138)
2160 RETURN
2170 REM  NACH LINKS
2180 IF ZP2=139 THEN 2220
2190 CALL SOUND(-1000,-4,5)
>
```

```
2200 CALL HCHAR(XP2,YP2,139)
2210 GOTO 2290
2220 CALL GCHAR(XP2,YP2-1,H2
)
2230 IF H2=144 THEN 2480
2240 CALL HCHAR(XP2,YP2,32)
2250 IF H2=152 THEN 2520
2260 YP2=YP2-1
2270 CALL SOUND(-1000,-4,5)
2280 CALL HCHAR(XP2,YP2,139)
2290 RETURN
2300 CALL SOUND(500,110,5)
2310 P1=P1-100
2320 GOSUB 4420
2330 RETURN
2340 CALL SOUND(400,-7,5)
2350 CALL HCHAR(XP1,YP1,133)
2360 CALL SOUND(600,-6,5)
2370 CALL HCHAR(XP1,YP1,134)
2380 CALL SOUND(800,-5,5)
2390 CALL HCHAR(XP1,YP1,135)
2400 CALL HCHAR(XP1,YP1,32)
2410 CALL HCHAR(XP2,YP2,32)
>
```

```
2420 P2=P2+1000
2430 PANZER1=PANZER1-1
2440 GOSUB 4490
2450 GOSUB 4560
2460 IF PANZER1<1 THEN 4620
2470 GOTO 700
2480 CALL SOUND(500,110,5)
2490 P2=P2-100
2500 GOSUB 4490
2510 RETURN
2520 CALL SOUND(400,-7,5)
2530 CALL HCHAR(XP2,YP2,141)
2540 CALL SOUND(600,-6,5)
2550 CALL HCHAR(XP2,YP2,142)
2560 CALL SOUND(800,-5,5)
2570 CALL HCHAR(XP2,YP2,143)
2580 CALL HCHAR(XP2,YP2,32)
2590 CALL HCHAR(XP1,YP1,32)
2600 P1=P1+1000
2610 PANZER2=PANZER2-1
2620 GOSUB 4420
2630 GOSUB 4590
2640 IF PANZER2<1 THEN 4620
>
```

```
2650 GOTO 700
2660 REM  PANZER 1 SCHUSS
2670 CALL GCHAR(XP1,YP1,ZP1)
2680 XK1=XP1
2690 YK1=YP1
2700 ON ZP1-127 GOSUB 2720,3
000,3180,3360
2710 RETURN
2720 REM  NACH OBEN
2730 XK1=XK1-1
2740 CALL GCHAR(XK1,YK1,ZK1)
2750 IF ZK1<>32 THEN 2800
2760 CALL HCHAR(XK1,YK1,132)
2770 CALL SOUND(100,-4,5)
2780 CALL HCHAR(XK1,YK1,32)
2790 GOTO 2720
2800 IF ZK1<>144 THEN 2850
```

```
2810 CALL SOUND(100,110,5)
2820 P1=P1-50
2830 GOSUB 4420
2840 GOTO 2710
2850 IF ZK1<>152 THEN 2880
2860 GOSUB 2900
>
```

```
2870 GOTO 2710
2880 GOSUB 2520
2890 GOTO 2710
2900 CALL SOUND(600,-5,5)
2910 CALL HCHAR(XK1,YK1,153)
2920 CALL SOUND(700,-6,5)
2930 CALL HCHAR(XK1,YK1,154)
2940 CALL SOUND(900,-7,5)
2950 CALL HCHAR(XK1,YK1,155)
2960 P1=P1-300
2970 GOSUB 4420
2980 CALL HCHAR(XK1,YK1,32)
2990 RETURN
3000 REM NACH RECHTS
3010 YK1=YK1+1
3020 CALL GCHAR(XK1,YK1,ZK1)
3030 IF ZK1<>32 THEN 3080
3040 CALL HCHAR(XK1,YK1,132)
3050 CALL SOUND(100,-4,5)
3060 CALL HCHAR(XK1,YK1,32)
3070 GOTO 3000
3080 IF ZK1<>144 THEN 3130
3090 CALL SOUND(100,110,5)
>
```

```
3100 P1=P1-50
3110 GOSUB 4420
3120 GOTO 2710
3130 IF ZK1<>152 THEN 3160
3140 GOSUB 2900
3150 GOTO 2710
3160 GOSUB 2520
3170 GOTO 2710
3180 REM NACH UNTEN
3190 XK1=XK1+1
3200 CALL GCHAR(XK1,YK1,ZK1)
3210 IF ZK1<>32 THEN 3260
3220 CALL HCHAR(XK1,YK1,132)
3230 CALL SOUND(100,-4,5)
3240 CALL HCHAR(XK1,YK1,32)
3250 GOTO 3180
3260 IF ZK1<>144 THEN 3310
3270 CALL SOUND(100,110,5)
3280 P1=P1-50
3290 GOSUB 4420
3300 GOTO 2710
3310 IF ZK1<>152 THEN 3340
3320 GOSUB 2900
>
```

```
3330 GOTO 2710
3340 GOSUB 2520
3350 GOTO 2710
3360 REM NACH LINKS
3370 YK1=YK1-1
3380 CALL GCHAR(XK1,YK1,ZK1)
3390 IF ZK1<>32 THEN 3440
3400 CALL HCHAR(XK1,YK1,132)
3410 CALL SOUND(100,-4,5)
3420 CALL HCHAR(XK1,YK1,32)
3430 GOTO 3360
3440 IF ZK1<>144 THEN 3310
3450 CALL SOUND(100,110,5)
3460 P1=P1-50
3470 GOSUB 4420
3480 GOTO 2710
3490 IF ZK1<>152 THEN 3520
3500 GOSUB 2900
3510 GOTO 2710
3520 GOSUB 2520
3530 GOTO 2710
3540 REM PANZER 2 SCHUSS
3550 CALL GCHAR(XP2,YP2,ZP2)
>
```

```
3560 XK2=XP2
3570 YK2=YP2
3580 ON ZP2-135 GOSUB 3600,3
780,3960,4140
3590 RETURN
3600 REM NACH OBEN
3610 XK2=XK2-1
3620 CALL GCHAR(XK2,YK2,ZK2)
3630 IF ZK2<>32 THEN 3680
3640 CALL HCHAR(XK2,YK2,140)
3650 CALL SOUND(100,-4,5)
3660 CALL HCHAR(XK2,YK2,32)
3670 GOTO 3600
3680 IF ZK2<>144 THEN 3730
3690 CALL SOUND(100,110,5)
3700 P2=P2-50
3710 GOSUB 4490
3720 GOTO 3590
3730 IF ZK2<>152 THEN 3760
3740 GOSUB 4320
3750 GOTO 3590
```

```
3760 GOSUB 2340
3770 GOTO 3590
>
```

```
3780 REM NACH RECHTS
3790 YK2=YK2+1
3800 CALL GCHAR(XK2,YK2,ZK2)
3810 IF ZK2<>32 THEN 3860
3820 CALL HCHAR(XK2,YK2,140)
3830 CALL SOUND(100,-4,5)
3840 CALL HCHAR(XK2,YK2,32)
3850 GOTO 3780
3860 IF ZK2<>144 THEN 3910
3870 CALL SOUND(100,110,5)
3880 P2=P2-50
3890 GOSUB 4490
3900 GOTO 3590
3910 IF ZK2<>152 THEN 3940
3920 GOSUB 4320
3930 GOTO 3590
3940 GOSUB 2340
3950 GOTO 3590
3960 REM NACH UNTEN
3970 XK2=XK2+1
3980 CALL GCHAR(XK2,YK2,ZK2)
3990 IF ZK2<>32 THEN 4040
4000 CALL HCHAR(XK2,YK2,140)
>
```

```
4010 CALL SOUND(100,-4,5)
4020 CALL HCHAR(XK2,YK2,32)
4030 GOTO 3960
4040 IF ZK2<>144 THEN 4090
4050 CALL SOUND(100,110,5)
4060 P2=P2-50
4070 GOSUB 4490
4080 GOTO 3590
4090 IF ZK2<>152 THEN 4120
4100 GOSUB 4320
4110 GOTO 3590
4120 GOSUB 2340
4130 GOTO 3590
4140 REM NACH LINKS
4150 YK2=YK2-1
4160 CALL GCHAR(XK2,YK2,ZK2)
4170 IF ZK2<>32 THEN 4220
4180 CALL HCHAR(XK2,YK2,140)
4190 CALL SOUND(100,-4,5)
4200 CALL HCHAR(XK2,YK2,32)
4210 GOTO 4140
4220 IF ZK2<>144 THEN 4270
4230 CALL SOUND(100,110,5)
>
```

```
4240 P2=P2-50
4250 GOSUB 4490
4260 GOTO 3590
4270 IF ZK2<>152 THEN 4300
4280 GOSUB 4320
4290 GOTO 3590
4300 GOSUB 2340
4310 GOTO 3590
4320 CALL SOUND(600,-5,5)
4330 CALL HCHAR(XK2,YK2,153)
4340 CALL SOUND(700,-6,5)
4350 CALL HCHAR(XK2,YK2,154)
4360 CALL SOUND(900,-7,5)
4370 CALL HCHAR(XK2,YK2,155)
4380 P2=P2-300
4390 GOSUB 4490
4400 CALL HCHAR(XK2,YK2,32)
4410 RETURN
4420 REM PUNKTE PANZER 1
4430 D1$=STR$(P1)&" P "
4440 FOR I=1 TO LEN(D1$)
4450 Z1=ASC(SEG$(D1$,I,1))
4460 CALL HCHAR(24,4+I,Z1)
>
```

```
>LIST 4470-4650
4470 NEXT I
4480 RETURN
4490 REM PUNKTE PANZER 2
4500 D2$=STR$(P2)&" P "
4510 FOR I=1 TO LEN(D2$)
4520 Z2=ASC(SEG$(D2$,I,1))
4530 CALL HCHAR(24,20+I,Z2)
4540 NEXT I
4550 RETURN
4560 PANZER1$=STR$(PANZER1)
4570 CALL HCHAR(24,2,ASC(PAN
ZER1$))
4580 RETURN
4590 PANZER2$=STR$(PANZER2)
4600 CALL HCHAR(24,18,ASC(PA
NZER2$))
4610 RETURN
4620 PRINT
4630 PRINT
4640 PRINT "SPIELENDE"
4650 PRINT
```

Fortsetzung auf Seite 61

TI-UFO

Ziel des Spieles ist es, ein Ufo um Hindernisse zu lenken. Die horizontale Steuerung erfolgt mit den Tasten A (nach links) und L (nach rechts). Wird keine Taste gedrückt, so bewegt sich das Ufo vertikal nach unten. Wird ein Hindernis senkrecht überfahren, so werden 2 Fehlerpunkte berechnet, beim waagrechten Überfahren 1 Fehlerpunkt. Ebenso werden beim waagrechten Zurückfahren und bei Erreichen des Bildschirmrandes Fehlerpunkte errechnet.

Die ersten drei Runden bestehen aus drei Durchgängen. Jede weitere Runde aus zwei Durchgängen. Der Schwierigkeitsgrad nimmt ständig zu. Werden in einer Runde weniger als 8 Fehlerpunkte verursacht, so kommt man in die nächste Runde (Freispiel). Der Schwierigkeitsgrad kann in Zeile 170, der Schwierigkeitszuwachs in der Zeile 770 verstellt werden.

Runden 1-3: # $ %
Runden 4 - 8: A B C D E
Runden 9 - : Zufallszeichen

Nachteil des Programms: Das Ufo kann aus Zeitgründen nicht gelöscht werden. Es bildet sich deshalb eine Ufoschlange.

Hans-Peter Adrion

```
100 CALL SCREEN(14)
110 CALL COLOR(1,16,14)
120 BU=1
130 JU=0
140 FE=0
150 ZE=35
160 REM HINDERNISSE PRO ZEIL
E
170 PIT=5
180 SC=0
190 FOR T=3 TO BU STEP -1
200 IF FE>=8 THEN 1050
210 CALL CLEAR
220 G=0
230 RANDOMIZE
240 GOSUB 560
250 IF FE>=8 THEN 1050
260 CALL SOUND(400,131,2,165
,2,196,2)
270 REM  **UFO SETZEN
280 SPAL=16
290 CALL CHAR(130,"000000183
C7E7E99")
300 G=G+1
310 CALL GCHAR(G,SPAL,WE)
320 IF WE>33 THEN 330 ELSE 3
60
330 CALL SOUND(100,-2,1)
340 CALL COLOR(1,5,14)
350 SC=SC+2
360 CALL HCHAR(G,SPAL,130)
370 IF G=24 THEN 660
380 CALL KEY(0,KEY,STATUS)
390 IF STATUS=0 THEN 300
400 IF KEY=65 THEN 460
410 IF KEY=76 THEN 430
420 GOTO 300
430 SPAL=SPAL+1
440 IF SPAL=33 THEN 460
450 GOTO 480
460 SPAL=SPAL-1
470 IF SPAL=0 THEN 430
480 CALL GCHAR(G,SPAL,Q)
490 IF Q>33 THEN 500 ELSE 53
0
500 CALL SOUND(100,-1,1)
510 CALL COLOR(1,16,14)
520 SC=SC+1
530 CALL HCHAR(G,SPAL,130)
540 GOTO 380
550 REM **HINDERNISSE SETZEN
560 Z=3
570 FOR M=1 TO 100
580 FOR C=1 TO PIT
590 X=INT(32*RND)+1
600 IF Z>24 THEN 650
610 CALL HCHAR(Z,X,ZE)
620 NEXT C
630 Z=Z+T
640 NEXT M
650 RETURN
660 CALL SOUND(2500,131,3,17
5,3,220,3)
670 NEXT T
680 CALL CLEAR
690 FE=FE+1
700 PRINT
710 REM **ASCII-CODE
720 PRINT
730 ZE=ZE+1
740 PRINT
750 PRINT
760 REM HINDERNISZUWACHS
770 PIT=PIT+4
780 REM AUSGABE
790 PRINT "FEHLERPUNKTE :";S
C
800 JU=JU+1
810 IF JU=3 THEN 1020
820 REM FREISPIEL SETZEN!
830 IF SC<=8 THEN 840 ELSE 9
30
840 PRINT
850 IF SC=0 THEN 860 ELSE 87
0
860 PRINT " O P T I M U M !"
870 PRINT
880 PRINT " ***FREISPIEL***"
890 REM PAUSE
900 FOR V=1 TO 800
910 NEXT V
920 GOTO 180
930 PRINT
940 PRINT "C BY HOMECOMPUTER
"
950 PRINT
960 PRINT "NOCH EIN SPIEL? (
J/N)"
970 CALL KEY(0,KEY,STATUS)
980 IF STATUS=0 THEN 970
990 IF KEY=74 THEN 120
1000 STOP
1010 REM ZEICHENSETZUNG AB
         RUNDE 8
1020 BU=2
1030 ZE=65
1040 GOTO 830
1050 CALL CLEAR
1060 TR=TR+50
1070 FOR OP=1 TO TR
1080 WA=INT(120*RND)+1
1090 WS=INT(22:RND)+3
1100 WD=INT(32*RND)+1
1110 CALL HCHAR(WS,WD,WA)
1120 NEXT OP
1130 G=0
1140 GOTO 260
1150 END
```

Killersatellit

Sobald das Spiel gestartet wurde, fragt der Computer nach der Wahl des Schiffes. Es stehen 4 Schiffe zur Wahl, die sich jedoch nur äußerlich unterscheiden. Die Steuerung des Schiffes erfolgt über die 4 Cursor-Tasten (E, S, D und X). Über die Space-Taste können Schüsse abgegeben werden. Nach jedem erfolgtem Abschuß erhöht sich die Schwierigkeit und Punktzahl der einzelnen Gegner.
Außerdem wird nach jedem Abschuß die eigene, sowie die Höchstpunktzahl angezeigt.
Das Spiel ist beendet, wenn der Satellit das Schiff abgeschossen hat. Alles Andere ergibt sich beim Spielen.

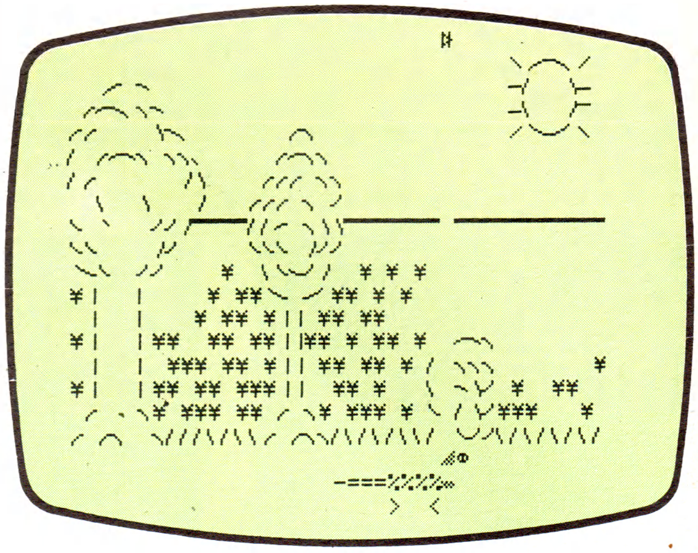
Ch. Bladoschewski

```
10 REM ********************
20 REM *  KILLERSATELLIT   *
30 REM *  ENTWICKELT VON   *
40 REM * CH. BLADOSCHEWSKI *
50 REM ********************
100 CALL CLEAR
110 CALL COLOR(1,2,2)
120 CALL SCREEN(2)
130 PRINT TAB(7);"KILLERSATE
LLIT": : : : : : : : : : : :
140 CALL CHAR(96,"00183C7E7E
E3C18")
150 CALL CHAR(104,"00183C7E7
E3C18")
160 CALL HCHAR(10,6,104,21)
170 CALL HCHAR(14,6,104,21)
180 CALL VCHAR(10,6,104,5)
190 CALL VCHAR(10,26,104,5)
200 FOR I=6 TO 26 STEP 2
210 CALL HCHAR(10,I,96)
220 CALL HCHAR(14,I,96)
230 NEXT I
240 CALL HCHAR(12,6,96)
250 CALL HCHAR(12,26,96)
260 CALL COLOR(1,16,1)
270 CALL SCREEN(16)
280 FOR I=1 TO 20
290 FOR J=1 TO 2
300 CALL COLOR(9,11,9)
310 CALL COLOR(10,5,13)
320 NEXT J
330 CALL COLOR(9,5,9)
340 CALL COLOR(10,11,13)
350 NEXT I
360 CALL COLOR(1,1,3)
370 CALL SCREEN(3)
380 REM
390 SC=0
400 CALL CLEAR
410 RANDOMIZE
420 M=2
430 P1=1
440 U=.1
450 A$="10101038BABAFE82"
460 R$="42425A665A4242"
470 C$="183C7E3C3C7EFF"
480 D$="82828200828282"
490 E$="1818181818181818"
500 B$="1818183C66FFFF66"
510 G$="42425A5A7EFFFF42"
520 INPUT "WELCHES SCHIFF WI
LLST DU (1-4)?":SCHIFF
530 IF (SCHIFF<1)+(SCHIFF>4)
<0 THEN 520
540 ON SCHIFF GOTO 640,580,6
10,550
550 Z$=R$
560 Y$=E$
570 GOTO 670
580 Z$=B$
590 Y$=E$
600 GOTO 670
610 Y$=D$
620 Z$=G$
630 GOTO 660
640 REM
650 Z$=A$
660 Y$=D$
670 X=10
680 Y=10
690 CALL CLEAR
700 B=12
710 A=16
720 CALL SCREEN(2)
730 CALL COLOR(1,1,1)
740 CALL COLOR(9,4,1)
750 CALL COLOR(10,14,1)
760 CALL COLOR(11,9,1)
770 CALL COLOR(12,5,1)
780 CALL CHAR(96,Y$)
790 CALL CHAR(104,"5A5A5A5A5
A5A5A5A")
800 CALL CHAR(112,C$)
810 CALL CHAR(120,Z$)
820 X=X-(INT((RND*2)-.5))
830 Y=Y-(INT((RND*2)-.5))
840 IF X<1 THEN 1240
850 IF X>24 THEN 1260
860 IF Y<1 THEN 1280
870 IF Y>32 THEN 1300
880 CALL KEY(0,K,S)
890 IF K=32 THEN 1320
900 IF K=83 THEN 1040
910 IF K=68 THEN 1070
920 IF K=88 THEN 1100
930 IF K=69 THEN 1130 ELSE 9
80
940 IF B<1 THEN 1160
950 IF B>24 THEN 1180
960 IF A<1 THEN 1200
970 IF A>32 THEN 1220
980 CALL CLEAR
990 CALL HCHAR(B,A,120)
1000 CALL HCHAR(X,Y,112)
1010 IF X>21 THEN 820
1020 IF RND<U THEN 1580
1030 GOTO 820
1040 A=A-1
1050 CALL SOUND(-500,-6,0)
1060 GOTO 940
1070 A=A+1
1080 CALL SOUND(-500,-6,0)
1090 GOTO 940
1100 B=B+1
1110 CALL SOUND(-500,-6,0)
1120 GOTO 940
1130 B=B-1
1140 CALL SOUND(-500,-6,0)
1150 GOTO 940
1160 B=B+24
1170 GOTO 980
1180 B=B-24
1190 GOTO 980
1200 A=A+32
1210 GOTO 980
1220 A=A-32
1230 GOTO 980
1240 X=X+24
1250 GOTO 880
1260 X=X-24
1270 GOTO 880
1280 Y=Y+32
1290 GOTO 880
1300 Y=Y-32
1310 GOTO880
1320 FOR P=B-1 TO B-M STEP -
1
1330 CALL SOUND(100,-5,0)
1340 CALL GCHAR(P,A,L)
1350 IF L=112 THEN 1390
1360 CALL HCHAR(P,A,96)
1370 NEXT P
1380 GOTO 980
1390 CALL CLEAR
1400 PRINT "GEGNER ZERSTOERT
": : : : :
1410 SC=SC+P1
1420 IF HS>=SC THEN 1490
1430 HS=SC
1440 FOR SI=1 TO 5
1450 FOR SJ=-3 TO -1
1460 CALL SOUND(100,SJ,0)
1470 NEXT SJ
1480 NEXT SI
1490 PRINT "HIGHSCORE:";HS
1500 PRINT "PUNKTZAHL:";SC:
: : : : : : : : : : :
1510 CALL COLOR(1,3,3)
1520 CALL SCREEN(3)
1530 P1=P1+1
1540 U=U+.02
1550 FOR I=1 TO 1000
1560 NEXT I
1570 GOTO 540
1580 FOR P=X+1 TO X+3
1590 CALL SOUND(100,-3,2)
1600 CALL GCHAR(P,Y,L)
1610 IF L=120 THEN 1650
1620 CALL HCHAR(P,Y,104)
1630 NEXT P
1640 GOTO 820
1650 CALL CLEAR
1660 PRINT TAB(5);"DU WURDES
T ZERSTOERT": : :"
GAME OVER": : :
1670 PRINT "HIGHSCORE:";HS
1680 PRINT "PUNKTZAHL:";SC:
: : : : : : :
1690 CALL COLOR(1,3,3)
1700 CALL SCREEN(3)
1710 PRINT "WILLST DU NOCH E
INMAL SPIE- LEN? DRUECKE <J>
 <N>"
1720 CALL KEY(0,K,S)
1730 IF S=0 THEN 1720
1740 IF K=ASC("J")THEN 380
1750 IF K<>ASC("N")THEN 1710
1760 END
```

Eidechse Sharp MZ-80K

Das Programm EIDECHSE wurde auf einem Sharp MZ-80 K geschrieben und ist ungefähr 5,5 KB lang. Die Grundidee ist einem Buch entnommen und wurde von mir erweitert und an den MZ-80 K angepaßt.
Zum Verlauf des Spieles sei gesagt, es ist ein sehr einfaches Spiel.
Der Spieler soll versuchen, mittels einer beliebigen Taste die Fliege zu fangen. Auf Tastendruck schnellt die Zunge der Eidechse nach oben und mit etwas Glück oder Geschick trifft er die Fliege.
Nach einer gewissen Trefferquote gibt es einen Zeitbonus. Die Zeit, Treffer und Durchgänge werden dem Spieler auf dem Bildschirm angezeigt.

Egon Kwapil



Die Eidechse mit der Fliege links von der Sonne

```
10 PRINT"EGON KWAPIL   HAMBURG 03.1983"
20 MUSIC"R7R7R7"
30 PRINT"            E I D E C H S E  ":MUSIC"R7"
40 PRINT"                                   ":MUSIC"R7"
50 PRINT"                                   ":MUSIC"R7"
60 PRINT"                                   ":MUSIC"R7"
70 PRINT"                                   ":MUSIC"R7"
80 PRINT"                                   ":MUSIC"R7"
90 PRINT"                                   ":MUSIC"R7"
100 PRINT"                                  ":MUSIC"R7"
110 PRINT"                                  ":FORA=0TO255:POKE4514,A
120 USR(68):NEXTA:PRINT"                              ":MUSIC"R7"
130 PRINT"                                     ":MUSIC"R7"
140 PRINT"                                     ":MUSIC"R7"
150 PRINT"                                     ":MUSIC"R7"
160 PRINT"                                     ":MUSIC"R7"
170 PRINT"                                     ":MUSIC"R7"
180 PRINT"                                     ":MUSIC"R7"
190 PRINT"                                  $  ":MUSIC"R7"
200 PRINT"                                     ":MUSIC"R7"
210 PRINT"                                     ":MUSIC"R72
220 PRINT"                                     ":MUSIC"R7"
230 PRINT"                                     ":MUSIC"R7"
240 PRINT"                                     ":MUSIC"R7"
250 PRINT"                                     ":MUSIC"R7"
260 A$="-CR3_DR3-ER3_DR3-DR3_CR3-DR3_ER3-CR3_DR3-ER3_ER3-DR3_DR3_C5"
270 B$="_CR3-DR3_ER3-DR3_DR3-CR3_DR3-ER3_CR3-DR3_ER3-ER3_DR3-DR3-C5"
280 C$="_CR3_DR3_ER3_DR3_DR3_CR3_DR3_ER3_CR3_DR3_ER3_ER3_DR3_DR3_C5"
290 TEMPO7:MUSICA$,B$,A$,C$
300 PRINT"                EIDECHSE"
310 PRINT"                ========="
```

```
320 REM▩ REAKTIONSSPIEL ▩
330 PRINT
340 PRINT"Die Eidechse versucht,die Fliege zu▩"
350 PRINT"Erhaschen.Zum Betätigen der Zunge▩"
360 PRINT"drücken Sie irgendeine Taste.▩"
370 PRINT"Sie haben dazu 3 Minuten Zeit.▩"
380 PRINT"Bei 10,20 oder 30 Punkten gibt es        Zeitbonus.▩"
390 PRINT"übrigens,die Fliege sieht so ▩ aus und"
400 PRINT"                              ▩ ↑ ▩              "
410 PRINT"ist schwer zu finden !!!      ↑ "
420 PRINT"                              ↑▩"
430 PRINT"          V I E L   S P A ß !"
440 PRINT"▩▩Drücken Sie jetzt eine Taste!"
450 GETT$:IFT$=""THEN450
460 REM▩ VORBEREITUNG ▩
470 PRINT"▩":MUSICC$
480 PRINTTAB(13);"▩▩▩▩▩▩▩▩▩▩Fertigmachen!"
490 REM
500 REM▩ VORWARNZEIT ▩
510 REM
520 FORI=1TO1000:NEXTI
530 REM
540 REM▩UHR WIRD GESTELLT ▩
550 REM
560 TI$="000000"
570 REM
580 REM▩ FLIEGENZÄHLER ▩
590 REM
600 Z=0:F=0
610 FORA=1TO200:POKE4514,A:USR(68):NEXT
620 REM
630 REM▩▩▩ SPIEL ▩▩▩
640 REM
650 PRINT"▩":F=F+1:REM▩ RUNDENZäHLER▩
660 POKE4466,5
670 PRINT"                                        \ ⌒ /"
680 PRINT"   ⌒ ⌒                                  -(   )-"
690 PRINT"  ⌒⌒ ⌒⌒                                 -(   )-"
700 PRINT"//      \⌒           ⌒                  / ⌒ \"
710 PRINT"/  ⌒⌒   \\          ⌒⌒          "
720 PRINT"/ /       /)       //⌒ ⌒\        "
730 PRINT"   \  \    /      //⌒ ⌒ )        "
740 PRINT"\  \   // /━━━━(( ⌒ ))━━━━━━━━━━━━━━━"
750 PRINT"\\       //       \\\ //         "
760 PRINT"⌒\   //       ¥   \ ⌒ /   ¥ ¥ ¥"
770 PRINT"¥|   |      ¥ ¥¥ ⌒ ⌒ ¥¥ ¥ ¥ "
780 PRINT" |   |    ¥ ¥¥ ¥|| ¥¥ ¥¥    "
790 PRINT"¥|   |¥¥  ¥¥ ¥¥||¥¥ ¥ ¥¥ ¥ ⌒⌒  "
800 PRINT" |   | ¥¥¥ ¥¥ ¥|| ¥¥ ¥¥ ¥ /⌒\\        ¥"
810 PRINT"¥|   |¥¥ ¥¥ ¥¥¥||  ¥¥ ¥   \/~Y ¥  ¥¥  "
820 PRINT" /   \¥ ¥¥¥ ¥¥ /\¥ ¥¥¥ ¥ Y/ Y¥¥¥    ¥ "
830 PRINT"/ ⌒   V//\/\\/ ⌒V/\/\/\\/\⌒/\/\/\/\/"
840 REM
850 REM▩ FLIEGE PLAZIEREN ▩
860 REM
870 R=(10* RND(1))
880 X=53328+11+R*20+(25*RND(1))
890 POKEX,231
900 REM
910 REM▩ EIDECHSE LAUFEN LASSEN ▩
920 REM
930 FORK=54168TO54206:REM▩KOPF VON EID.▩
940 POKEK-1,0:POKEK,0:POKEK-41,0:POKEK-40,0
950 POKEK,212:POKEK-1,101:POKEK-2,101:POKEK-3,101
960 POKEK-4,101:POKEK-5,43:POKEK-6,43:POKEK-7,43
970 POKEK-8,42:POKEK-40,216:POKEK-39,191:POKEK-9,0
980 POKEK+39,81:POKEK+38,0:POKEK+36,87:POKEK+35,0
990 REM
1000 REM▩ ZEITANSAGE ▩
1010 REM
1020 POKE4466,1:PRINT"ZEIT:";TI$;" ▩"
1030 POKE4466,1:POKE4465,15:PRINT"FLIEGEN :";Z;" ▩ VERSUCH";F
1040 TI=VAL(TI$)
1050 IFTI>000300THEN1550
1060 GETA$
1070 IFA$=""THEN1110:REM▩ NÄCHSTER SCHRITT DER EIDECHSE▩
1080 GOSUB1140:REM▩ FANGVERSUCH ▩
1090 IFTI<000300THEN650:REM▩ AUF EIN NEUES ▩
1100 GOTO1550
1110 NEXTK
1120 GOTO650:REM▩ NEUER VERSUCHCH ▩
1130 REM
1140 REM▩▩▩ UNTERPGM.FANGVERSUCH ▩▩▩
1150 REM
1160 FORT=K-80 TOK-800STEP-40
1170 POKET,238
1180 FORM=1TO30STEP+9 :POKE4514,M:USR(68)
1190 IFPEEK(T-40)<>231THEN1340:REM▩ NICHTS ERWISCHT ▩
1200 FORH=TTOKSTEP40:REM▩ BEUTE EINROLLEN ▩
1210 POKEH-40,0:POKEH,85
1220 FORI=1TO5:NEXTI
1230 NEXTH
1240 REM
1250 REM▩ BEUTE VERSCHLINGEN ▩
1260 REM
1270 POKEK-39,0:POKEK-40,0:POKEK,209
1280 POKEK+1,214:POKEK+2,191
1290 FORI=1TO50:NEXTI
1300 TEMPO7:P=54149:FORN=PTOP-400STEP-40:MUSIC"D0F0G0":NEXTN:MUSIC"‾C0‾D0‾E0‾"
1310 FORV=1TO20:POKEK,209:POKEK+1,214:POKEK+1,208:POKEK+2,191:NEXTV
1320 GOSUB1370:REM▩ SCHLUCKEN▩
1330 POKET,238:GOSUB1450:RETURN
1340 NEXTM:NEXTT
1350 RETURN
1360 REM
1370 REM▩ UNTERPGM.SCHLUCKEN ▩
1380 REM
1390 PRINT:PRINT"SCHMATZ!!"
1400 Z=Z+1
1410 PRINT:PRINTZ;" FLIEGEN GESCHNAPPT"
1420 FORI=1TO1000:NEXTI:REM▩ VERDAUUNGSPAUSE▩
1430 RETURN
1440 REM
1450 REM▩ BONUSZEIT ▩
1460 REM
1470 IFZ=10THENPRINT"▩B▩O▩N▩U▩S▩Z▩E▩I▩T !":TI$="000030":GOSUB1510:GOTO630
1480 IFZ=20THENPRINT"▩B▩O▩N▩U▩S▩Z▩E▩I▩T !":TI$="000050":GOSUB1510:GOTO630
1490 IFZ=30THENPRINT"▩B▩O▩N▩U▩S▩Z▩E▩I▩T !":TI$="000110":GOSUB1510:GOTO630
1500 RETURN
1510 TEMPO1:MUSIC"‾D‾F‾G‾A‾F":FORL=1TO500:NEXTL:RETURN
1520 REM
1530 REM▩▩▩ SPIELENDE UND AUSWERTUNG ▩▩▩
1540 REM
1550 PRINT"▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩ DIE ZEIT IST UM !!!!!!!!!▩▩▩▩▩▩▩"
1560 Q=(F-Z)/(F)*100-100:Q=Q-Q-Q
1570 PRINT"Sie haben in genau ";F;" Versuchen▩▩▩▩▩▩▩ ";Z;
1580 PRINT" Fliegen gefangen.▩▩"
1590 PRINT"Das ergibt eine Fangrate von :▩";Q;" %"
1600 IFZ<7THEN1630
1610 IFZ=7THEN1670
1620 IFZ>7THEN1700
1630 PRINT"▩▩▩▩▩Etwas wenig,finden Sie nicht auch ? Das Tapfere";
1640 PRINT" Schneiderlein schaffte 7 auf     einen Streich.Etwas Übung könnte";
1650 PRINT" Ihnen  nicht schaden !!!!!"
1660 GOTO1720
1670 PRINT"▩▩▩▩▩Bravo genau so gut wie das Tapfere       Schneiderlein,";
1680 PRINT" aber bei dem Technischen Stand von Heute ist das ja keine allzu";
1690 PRINT"  große Leistung.":GOTO1720
1700 PRINT"▩▩▩▩▩Bravo Sie sind hiermit zum neuen Tap-    feren Schneiderlein";
1710 PRINT" der Neuzeit ernannt.":GOTO1720
1720 PRINT"▩▩ Noch einmal ? (J/N)"
1730 GETN$:IFN$=""THEN1730
1740 IFN$="J"THEN470
1750 IFN$="N"THEN1770
1760 GOTO1730
1770 PRINT"▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩D A N K E   F Ü R   D A S   S P I E L .▩▩▩▩▩▩▩▩▩"
1780 END
```

Chiffrier/ Dechiffrier-Programm

TI 99/4A + Drucker

Eingabe
Das Pgm fragt zunächst, ob dechiffriert werden soll, – dann nur DE eingeben (wird im Pgm auch gesagt) Ansonsten verlangt es die Eingabe einer vierstelligen Codezahl (z.B. Datum in der Form 3103 = 31.03.) Diese Codezahl muß auch dem Entschlüssler bekannt sein.
Das Pgm nimmt nur Zahlen an und zwar nur eine! Es sind also alle vier einzeln einzugeben. Danach wird die Eingabe einer zu verschlüsselnden Zeile verlangt. Das Pgm nimmt nur 28 Zeichen an und sagt an, daß der Zeilenrest ggflls mit Punkten zu füllen ist. Die Eingabe geschieht ganz normal und der Drucker druckt sie verschlüsselt aus. So kann Zeile für Zeile eingegeben werden. Die letzte Eingabe soll eine Ø sein, damit das Pgm erkennt, daß die Eingabe abgeschlossen ist.
Das Dechiffrieren geschieht genauso, nur daß hier eben der chiffrierte Text eingegeben wird, der dann Zeile für Zeile dechiffriert herauskommt.
Interessant und daher kaum zu „knacken" ist der Algorhitmus des Pgms hinsichtlich der Verschlüsselung. Manchmal ist der gleiche Buchstabe von gleicher Bedeutung, manchmal haben 2 gleiche Buchstaben eine völlig andere Bedeutung und manchmal werden für einen Buchstaben im laufe eines längeren Textes bis zu 50 andere eingesetzt.

Programmbeispiel:
Eingegeben wurde zeilenweise

DIES IST EIN CHIFFRIERTEST:: AUFGENOMMEN AM 31.03.83 FÜR DIE ZEITSCHRIFT HOMECOMPUTER

Da die Eingabe automatisch auf 28 Zeichen beschränkt ist, können Sie am Zeilenende jeweils auf dem . stehenbleiben, der Computer meldet sich mit einem BEEP.
So sah der Ausdruck aus:

```
AHEP HSQ DIK BHFCERFBQTBPS.+
>TFDBMOJJDN >L 0.-00+73 CTED
AHE WDIQPBHOFET ENMB@NMMRSED
```

und als der vorige Ausdruck Zeichen für Zeichen, wiederum zeilenweise nach vorheriger Eingabe der Codeziffern eingegeben wurde, kam folgender Ausdruck zustande:

```
DIES IST EIN CHIFFRIERTEST..
AUFGENOMMEN AM 31.03.83 FUER
DIE ZEITSCHRIFT HOMECOMPUTER
```

```
100 CALL CLEAR :: CALL SCREEN(16
)
110 DIM B$(30)
120 DIM B(30)
130 INPUT "ZUM DECHIFFRIEREN DE
EINGEBEN":DE$ :: IF DE$="DE" THE
N 370
140 PRINT "GIB JETZT EINZELN DIE
 4 CODEZIFFERN EIN:"
150 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):W
160 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):X
170 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Y
180 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Z
190 PRINT "TEXTZEILE EINGEBEN  (
BIS 28.STELLE MIT PUNKTEN FUELLE
N):"
200 ACCEPT BEEP SIZE(28):A$
210 IF A$="0" THEN END
220 FOR I=1 TO 28
230 B(I)=ASC(SEG$(A$,I,1))
240 NEXT I
250 FOR I=1 TO 28 STEP 4 :: B(I)
=B(I)-W :: NEXT I
260 FOR I=2 TO 28 STEP 4 :: B(I)
=B(I)-X :: NEXT I
270 FOR I=3 TO 28 STEP 4 :: B(I)
=B(I)-Y :: NEXT I
280 FOR I=4 TO 28 STEP 4 :: B(I)
=B(I)-Z :: NEXT I
290 FOR I=1 TO 28
300 IF B(I)<=32 THEN B(I)=32
310 NEXT I
320 FOR I=1 TO 28 :: B$(I)=CHR$(
B(I)):: NEXT I
330 OPEN #1:"TP.E",OUTPUT
340 FOR I=1 TO 28 :: PRINT #1:B$
(I);:: NEXT I
350 CLOSE #1
360 GOTO 200
370 PRINT "GIB JETZT EINZELN DIE
 4 CODEZIFFERN EIN:"
380 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):W
390 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):X
400 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Y
410 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Z
420 PRINT "GIB DEN ZEILENTEXT EI
N":A$
430 ACCEPT BEEP SIZE(28):A$
440 IF A$="0" THEN END
450 FOR I=1 TO 28
460 B(I)=ASC(SEG$(A$,I,1))
470 NEXT I
480 FOR I=1 TO 28 STEP 4 :: B(I)
=B(I)+W :: NEXT I
490 FOR I=2 TO 28 STEP 4 :: B(I)
=B(I)+X :: NEXT I
```

```
500 FOR I=3 TO 28 STEP 4 :: B(I)
=B(I)+Y :: NEXT I
510 FOR I=4 TO 28 STEP 4 :: B(I)
=B(I)+Z :: NEXT I
520 FOR I=1 TO 28
530 IF B(I)<=41 THEN B(I)=32
540 NEXT I
550 FOR I=1 TO 28 :: B$(I)=CHR$(
B(I)):: NEXT I
560 OPEN #1:"TP.E",OUTPUT
570 FOR I=1 TO 28 :: PRINT #1:B$
(I);:: NEXT I
580 CLOSE #1
590 GOTO 430
600 END
```

Rangliste

TI 99/4A + Drucker

Aufgrund einer Wahl oder eines Wettbewerbes soll im Verein z. B. eine Rangliste der Ergebnisse aufgestellt werden.

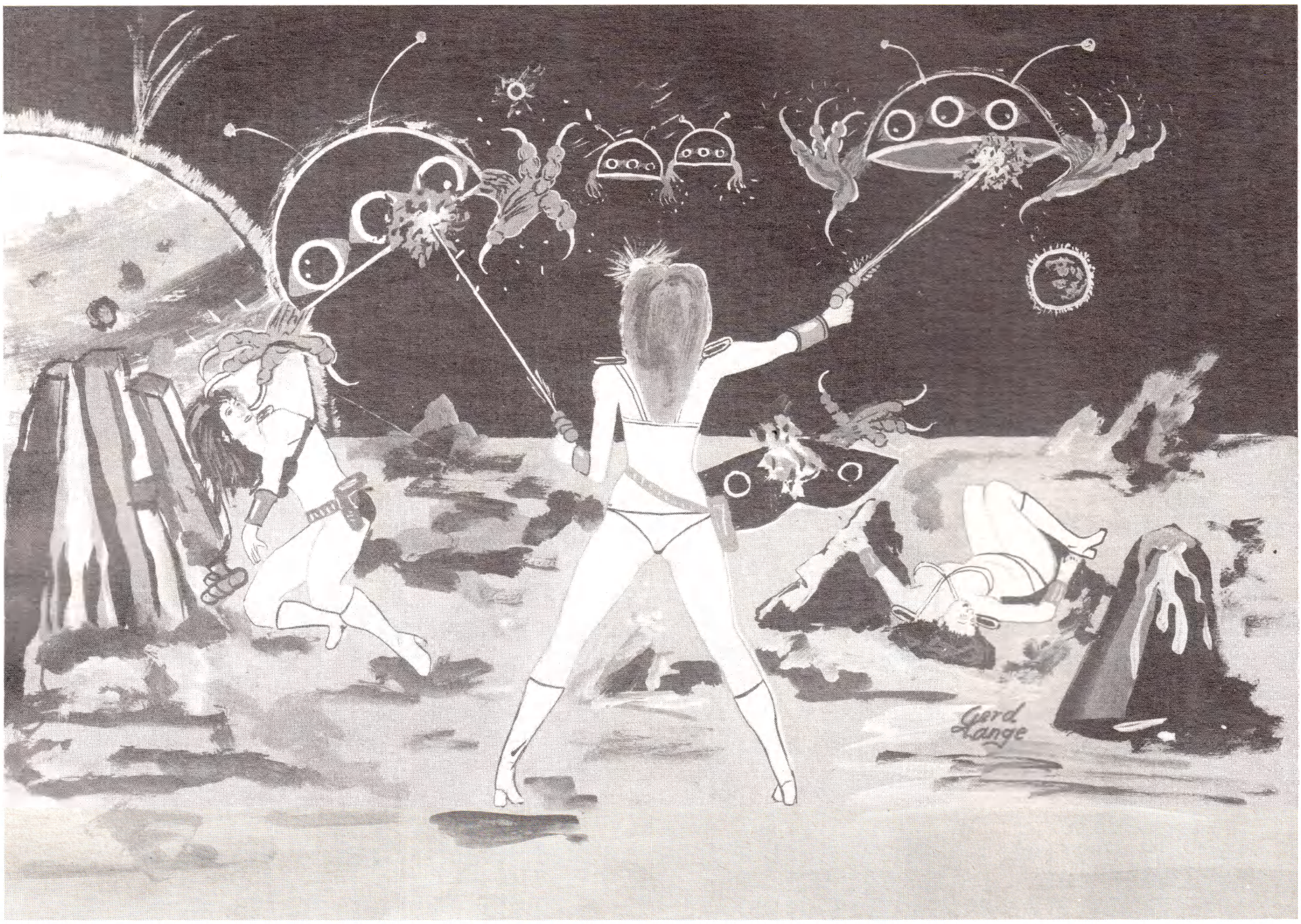
Die Verarbeitungszeit für 100 Teilnehmer beträgt ca. 5 min.

Das Programm fragt in einer Dreiergruppe für jeden Teilnehmer:
LISTENPLATZ
NAME
ERGEBNIS (in Punkten oder Stimmen)
Sind die Angaben für alle Teilnehmer eingegeben, erwartet das Pgm zum Abschluß unter LISTENPLATZ die Eingabe einer Ø und beginnt die Verarbeitung.

Ausgabeform:
1. (5.) Mayerhofer = 189
dabei ist die erste Zahl der Ranglistenplatz die in ( ) stehende der bisherige Listenplatz und nach dem Namen folgt das Ergebnis aus den Punkten/Stimmen.
Das Pgm stoppt – zur Verlesung der Ergebnisse – automatisch nach der Ausgabe von jeweils 5 Ergebnissen und kann dann durch Drücken irgendeiner Taste fortgesetzt werden.
Es erfolgt die Ausgabe eines kompletten Druckprotokolls.
Falls mehr als 99 Teilnehmer erfaßt werden sollen, sind die DIM-Statements in den Zeilen 110 - 120 - 130 entsprechend der Teilnehmerzahl + 1 zu erhöhen.
Das Programm läuft auf dem TI 99/4 A mit Extended-Basic-Modul + Drukker.

Herbert May

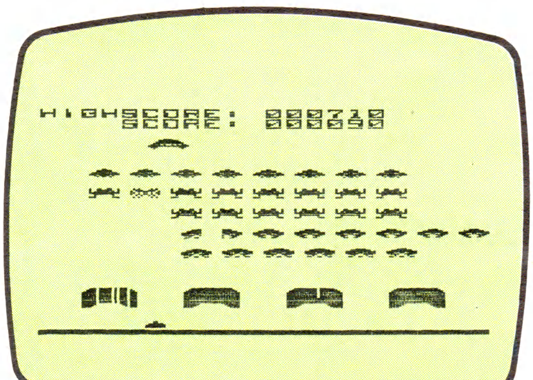
```
1 CALL SCREEN(16)
2 DIM LP(100)
100 DIM CY$(100)
110 DIM A(100)
120 FOR I=1 TO 100
121 PRINT
122 INPUT "LISTENPLATZ      ":LP(I
)
123 IF LP(I)=0 THEN 220
124 INPUT "NAME             ":CY$(
I)
125 INPUT "ERGEBNIS         ":A(I)
126 N=N+1
127 LP(I)=LP(I)/100
128 A(I)=A(I)+LP(I)
129 NEXT I
220 GOSUB 320
221 CALL CLEAR
230 FOR I=1 TO 100
240 AA=ABS(A(I)-INT(A(I)))
250 A(I)=INT(A(I))
260 AA=AA*100
270 B=B+1
275 PRINT : :
280 PRINT :B;".(";AA;")";TAB(12)
;CY$(AA);":";TAB(25);A(I)
285 OPEN #1:"TP.E",OUTPUT :: PRI
NT #1:B;".(";AA;")";TAB(12);CY$(
AA);":";TAB(25);A(I):: CLOSE #1
290 Z=Z+1
300 IF Z=5 THEN 590
301 IF AA=0 THEN 311
310 NEXT I
311 END
320 HL=N
330 FOR HK=INT(N/2)TO 1 STEP -1
340 HI=HK
350 GOSUB 460
360 NEXT HK
370 FOR HK=N TO 2 STEP -1
380 HH=A(1)
390 A(1)=A(HK)
400 A(HK)=HH
410 HI=1
420 HL=HK-1
430 GOSUB 460
440 NEXT HK
450 RETURN
460 HJ=2*HI
470 IF HJ>HL THEN 570
480 IF HJ=HL THEN 510
490 IF A(HJ)<A(HJ+1)THEN 510
500 HJ=HJ+1
510 IF A(HJ)>A(HI)THEN 570
520 HH=A(HI)
530 A(HI)=A(HJ)
540 A(HJ)=HH
550 HI=HJ
560 GOTO 460
570 RETURN
580 Z=Z+1
590 Z=0
600 GOSUB 620
610 GOTO 310
620 CALL KEY(0,J,ST)
630 IF ST=0 THEN 620
640 IF J=74 THEN 650
650 RETURN
660 Z=0
```



HC-Invaders

Zum ersten populären Computer-Videospiel wurden weltweit die Space-Invaders, kleine Monster aus dem All, die versuchen, die Erde zu erobern.
Sie haben es geschafft; keine Frage!

Homecomputer bringt Ihnen für Ihren VC-20, damit Sie Gelegenheit erhalten, den Editor/Assembler zu testen, eine eigene Version der bekannten Invaders ins Haus.



Instruktionen:

1. Editor eingeladen
2. INVADERS (Quellprogramm) eingeben
3. Assembler einladen
4. POKE 55,254: POKE 56,55: RUN
5. POKE 43,254: POKE 44,55: POKE 45,64: POKE 46,63: POKE 56,64: POKE 55,0
6. SAVE „INVADERS“ (MASCHINENPROGRAMM)
7. POKE 43,1: POKE 44,28: POKE 168,0:
8. LOAD „ZEICHENDATAS“
9. RUN
10. LOAD „INVADERS“ (BASIC-PGM)
11. RUN

Wenn alles auf Kassette gespeichert ist, geht man vor wie folgt: POKE 43,254: POKE 44,55: „LOAD“ (Maschinenprogramm).
LOAD „INVADERS“ (Maschinenprogramm) fahre fort wie oben ab 7.

ZEICHENDATAS

```
10 FORI=5120TO6511:READE:POKEI,E:NEXT
1000 DATA124,64,124,68,124,12,12,124,60,36,126,98,98,98,98,0
1010 DATA124,36,62,50,50,50,126,0,126,66,64,96,96,98,126,0
1020 DATA126,34,34,50,50,50,126,0,126,64,64,120,96,96,126,0
1030 DATA126,64,64,120,96,96,96,0,126,66,64,110,98,98,126,0
1040 DATA66,66,66,126,98,98,98,0,16,16,16,24,24,24,24,0
1050 DATA2,2,2,6,6,70,60,0,66,68,72,126,98,98,98,0
1060 DATA64,64,64,96,96,96,126,0,102,90,66,98,98,98,98,0
1070 DATA114,74,74,106,106,106,102,0,126,66,66,98,98,98,126,0
1080 DATA126,66,66,126,96,96,96,0,126,66,66,98,106,100,122,0
1090 DATA126,66,66,126,100,98,98,0,126,66,64,126,6,70,126,0
1100 DATA124,16,16,24,24,24,24,0,66,66,66,98,98,98,126,0
1110 DATA66,66,66,36,36,24,24,0,98,98,98,98,66,90,102,0
1120 DATA66,36,24,124,98,98,98,0,66,66,66,60,24,24,24,0
1130 DATA126,2,28,96,96,96,126,0,60,32,32,48,48,48,60,0
1140 DATA60,126,255,255,255,255,126,60,60,4,4,12,12,12,60,0
1150 DATA0,8,28,42,8,8,8,8,0,0,16,32,127,32,16,0
1160 DATA0,0,0,0,0,0,0,0,24,24,28,28,28,0,28,0
1170 DATA108,108,36,36,0,0,0,0,36,36,126,52,126,52,52,0
1180 DATA8,62,40,62,26,62,24,0,112,82,116,8,30,42,78,0
1190 DATA120,72,48,112,106,100,122,0,56,56,8,8,0,0,0,0
1200 DATA12,16,32,48,48,16,12,0,48,8,4,12,12,8,48,0
1210 DATA8,42,28,62,28,42,8,0,0,8,8,62,62,12,12,0
1220 DATA0,0,0,0,56,56,8,8,0,0,0,126,126,0,0,0
1230 DATA0,0,0,0,0,24,24,0,0,2,4,8,16,32,64,0
1240 DATA126,66,70,106,114,98,126,0,56,8,8,24,24,24,60,0
1250 DATA126,66,2,126,96,96,126,0,62,2,2,30,6,6,126,0
1260 DATA64,64,68,68,126,12,12,0,62,32,32,62,6,70,126,0
1270 DATA126,64,64,126,98,98,126,0,126,66,4,8,24,24,24,0
1280 DATA60,36,36,126,98,98,126,0,126,66,66,126,6,6,126,0
1290 DATA0,0,24,0,24,24,0,0,0,0,56,0,56,56,8,8
1300 DATA14,24,48,96,48,24,14,0,0,0,126,0,126,126,0,0
1310 DATA112,24,12,6,12,24,112,0,126,66,2,62,48,0,48,0
1320 DATA6,31,31,102,255,47,198,0,0,128,128,96,240,64,48,0
1330 DATA0,1,1,6,15,2,1,0,96,248,248,102,255,244,104,0
1340 DATA25,63,47,255,191,160,153,0,128,128,64,240,208,80,144,0
1350 DATA9,9,10,15,3,2,14,0,153,249,245,255,252,4,7,0
1360 DATA15,63,255,230,121,16,102,0,0,192,240,112,224,128,96,0
1370 DATA0,3,15,14,7,3,1,0,240,252,255,103,158,12,152,0
1380 DATA0,0,24,24,126,255,255,255,0,0,0,0,0,0,0,0
1390 DATA0,0,6,6,31,63,63,63,0,0,0,0,128,192,192,192
1400 DATA0,0,1,1,7,15,15,15,0,0,128,128,224,240,240,240
1410 DATA0,0,0,0,1,3,3,3,0,0,96,96,248,252,252,252
1420 DATA3,15,31,63,106,255,112,32,192,240,248,252,86,255,14,4
1430 DATA0,0,0,0,0,0,0,0,1,7,15,31,53,127,56,16
1440 DATA224,248,252,254,43,255,7,2,0,0,0,0,0,128,0,0
1450 DATA0,3,7,15,26,63,28,8,240,252,254,255,149,255,3,1
1460 DATA0,0,0,0,128,192,128,0,0,1,3,7,13,31,14,4
1470 DATA120,254,255,255,74,255,1,0,0,0,0,128,192,224,192,128
1480 DATA0,0,1,3,6,15,7,2,60,255,255,255,165,255,0,0
1490 DATA0,0,128,192,96,240,224,64,0,0,0,1,3,7,3,1
1500 DATA30,127,255,255,82,255,128,0,0,128,192,224,176,248,112,32
1510 DATA0,0,0,0,1,3,1,0,15,63,127,255,169,255,192,128
1520 DATA0,192,224,240,88,252,56,16,0,0,0,0,0,1,0,0
1530 DATA7,31,63,127,212,255,224,64,128,224,240,248,172,254,28,8
1540 DATA80,41,150,64,150,41,80,0,160,64,144,32,144,64,160,0
1550 DATA5,2,9,4,9,2,5,0,10,148,105,2,105,148,10,0
1560 DATA124,254,127,181,33,0,0,0,0,0,0,0,0,0,0,0
1570 DATA31,63,31,45,8,0,0,0,0,128,192,64,64,0,0,0
1580 DATA7,15,7,11,2,0,0,0,192,224,240,80,16,0,0,0
1590 DATA1,3,1,2,0,0,0,0,240,248,252,212,132,0,0,0
1600 DATA164,85,171,106,60,126,126,60,0,0,0,0,0,0,0,0
1610 DATA10,5,10,6,3,7,7,3,64,80,176,160,192,224,224,192
1620 DATA144,34,85,40,44,58,127,255,0,0,0,0,0,0,0,0
1630 DATA36,8,21,10,11,14,31,63,0,128,64,0,0,0,192,192
1640 DATA9,2,5,2,2,3,7,15,0,32,80,128,192,160,240,240
1650 DATA2,0,1,0,0,0,1,3,64,136,84,160,176,232,252,252
1660 DATA9,68,170,20,52,126,254,255,0,0,0,0,0,0,0,0
1670 DATA2,17,42,5,13,23,63,63,64,0,128,0,0,0,128,192
1680 DATA0,4,10,1,3,5,15,15,144,64,160,64,64,192,224,240
1690 DATA0,1,2,0,0,1,3,3,36,16,168,80,208,112,248,252
1700 DATA15,31,63,127,127,127,127,127,255,255,255,255,255,255,255,255
1710 DATA240,248,252,254,254,254,254,254,15,31,63,127,127,127,127,127
1720 DATA255,255,255,255,255,255,255,255,240,248,252,254,254,254,254,254
1730 DATA15,31,63,127,127,127,127,127,255,255,255,255,255,255,255,255
1740 DATA240,248,252,254,254,254,254,254,15,31,63,127,127,127,127,127
1750 DATA255,255,255,255,255,255,255,255,240,248,252,254,254,254,254,254
1760 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1770 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1780 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1790 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1800 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1810 DATA127,127,127,127,127,126,124,120,255,255,255,255,0,0,0,0
1820 DATA254,254,254,254,254,126,62,30,127,127,127,127,127,126,124,120
1830 DATA255,255,255,255,0,0,0,0,254,254,254,254,254,126,62,30
1840 DATA127,127,127,127,127,126,124,120,255,255,255,255,0,0,0,0
1850 DATA254,254,254,254,254,126,62,30,127,127,127,127,127,126,124,120
1860 DATA255,255,255,255,0,0,0,0,254,254,254,254,254,126,62,30

READY.
```

INVADERS (BASIC-PGM)

```
50 POKE55,254:POKE56,55
100 POKE5120,0:POKE5121,0:FORI=5122TO5127:POKEI,255:NEXT:REM BODEN IM CHR-RAM
110 POKE15841,0:REM NEUER BODEN
120 POKE15752,2:REMHIMMELBLINKGESCHWINDIGKEIT
130 POKE15806,3:REM BODENBLINKGESCHWINDIGKEIT
140 POKE14812,15:REMBODENBLINKANZAHL
150 POKE14916,4:REM HIMMELBLINKANZAHL
160 POKE16190,201:POKE16191,64:POKE16192,144:POKE16193,5:POKE16194,201:POKE16195
,76
170 POKE16196,176:POKE16197,1:POKE16198,96:POKE16199,104:POKE16200,24:POKE16201,
105
180 POKE16202,4:POKE16203,72:POKE16204,96
190 POKE15429,234:POKE15430,32:POKE15431,62:POKE15432,63
200 POKE15513,234:POKE15514,32:POKE15515,62:POKE15516,63
210 POKE15351,2:REM INVADERGESCHWINDIGKEIT
220 POKE16018,66:REM BLAUE FARBE
230 POKE15031,5:REM INVADERBLINKANZAHL
240 GOTO270 SCHUSSKORREKTUR
250 S=0:FORI=4136TOI-5STEP-1:S=10*S+(15ANDPEEK(I)):NEXT
260 IFS>HITHENHI=S
270 FORI=37888+66TOI+21:POKEI,5:NEXT
280 FORI=37954TOI+21:POKEI,53:NEXT
290 POKE36869,205
300 PRINT"▒▒HIGHSCORE: "RIGHT$(STR$(1E6+HI),6)
310 PRINT"     SCORE: 000000"
320 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒@@@@@@@@@@@@@@@@@@@@@@";:POKE4601,0:POKE38393,5
330 POKE912,1:REM ANZAHL BASEN
340 FORI=37955TOI+19:POKEI,4:NEXT
350 S1=15339:REM INVADER-MOVE
360 S2=15239:REM UFO
370 S3=15103:REM TASTATURABFRAGE
380 S4=14853:REM SPIELERSCHUSS WEITERBEWEGEN
390 S5=15740:REM HIMMELBLINKEN
400 S6=15794:REM BODENBLINKEN
410 S7=15848:REM INVADERBLINKEN
420 S8=14526:REM INVADERSCHUSS INITIALISIEREN UND WEITERBEWEGEN
430 S9=15669:REM BASISBLINKEN
440 FORI=832TO860:POKEI,0:NEXT
450 FORI=864TOI+44:POKEI,256*RND(1):NEXT
460 POKE862,88:POKE863,16:POKE839,88:POKE840,16:POKE36869,205
470 POKE851,16:POKE852,4:POKE853,1:POKE854,64
480 SYS15936:POKE3,0
490 FORI=37888+21*22TOI+21:POKEI,2:NEXT
500 IFPEEK(3)THEN250
510 IFPEEK(861)>AITHENGOSUB550
520 AI=PEEK(861)
530 SYSS1:SYSS2:SYSS3:SYSS4:SYSS5:SYSS6:SYSS7:SYSS8:SYSS9
540 GOTO500
550 RESTORE:FORI=6240TOI+23:READR:POKEI,R:POKEI+24,R:POKEI+48,R:POKEI+72,R:NEXT
560 FORI=6416TOI+23:READR:POKEI,R:POKEI+24,R:POKEI+48,R:POKEI+72,R:NEXT
570 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒LMN▒▒OPQ▒▒RST▒▒UVW"
580 PRINT"▒▒▒"CHR$(34);:POKE212,0:PRINT"#$▒  ▒%&'▒  ▒()*▒  ▒+,-
590 RETURN
600 DATA15,31,63,127,127,127,127,127,255,255,255,255,255,255,255,255
610 DATA240,248,252,254,254,254,254,254
620 DATA127,127,127,127,127,126,124,120,255,255,255,255,0,0,0,0
630 DATA254,254,254,254,254,126,62,30
```

Starship VC-20

INVADERS (Quellprogramm)

```
  1                ORG $37FE
  2PTR1            EQU $33C
  3PTR2            EQU PTR1+1
  4PTR3            EQU PTR2+1
  5PTR4            EQU PTR3+1
  6UFF             EQU $340
  7UP              EQU $341
  8BOBZ            EQU $342
  9BBZ             EQU $343
 10ISF             EQU $344
 11ZUISCH          EQU $345
 12ZUSSCH          EQU $346
 13INVI            EQU $347
 14ERG             EQU $351
 15BP              EQU $352
 16TAB             EQU $353
 17SSF             EQU $357
 18ISCHADR         EQU 0
 19SSCHADR         EQU 193
 20HIM             EQU $358
 21HIMZ            EQU $359
 22SBDZ            EQU $35A
 23BI              EQU $35B
 24IBZ             EQU $35C
 25AZI             EQU $35D
 26INVA            EQU $35E
 27AZB             EQU $390
 28LIZ             EQU $391
 29MR              EQU $34C
 30BDZ             EQU $34D
 31MDZ             EQU $34E
 32SC              EQU 4128
 33ULA             EQU 4096+66
 34BLA             EQU 4558
 35F1              EQU 5
 36F2              EQU 2
 37F3              EQU 3
 38                BRK
 39                BRK
 40RND             LDX PTR1
 41                LDA $360,X
 42                LDX PTR2
 43                ADC $36D,X
 44                LDX PTR3
 45                ADC $37C,X
 46                LDX PTR4
 47                STA $360,X
 48                PHA
 49                LDY #0
 50                INC PTR1
 51                LDA PTR1
 52                CMP #13
 53                BNE *+3
 54                STY PTR1
 55                INC PTR2
 56                LDA PTR2
 57                CMP #15
 58                BNE *+3
 59                STY PTR2
 60                INC PTR3
 61                LDA PTR3
 62                CMP #17
 63                BNE *+3
 64                STY PTR3
 65                INC PTR4
 66                LDA PTR4
 67                CMP #45
 68                BNE *+3
 69                STY PTR4
 70                PLA
 71                RTS
 72CHRRADR         LDY #0
 73                STY $FF
 74                ASL
 75                ROL $FF
 76                ASL
 77                ROL $FF
 78                ASL
 79                ROL $FF
 80                STA $FE
 81                CLC
 82                LDA $FF
 83                ADC #20
 84                STA $FF
 85                RTS
 86SC0             STA 87
 87                SED
 88                CLC
 89                LDX #4
 90                LDY #0
 91SC1             ADC 87
 92                BCC *+2
 93                INY
 94                CLC
 95                DEX
 96                BNE SC1
 97                STA 87
 98                CLD
 99                AND #15
100                ADC SC+6
101                STA SC+6
102                CMP #$3A
103                BCC *+5
104                SBC #10
105                STA SC+6
106                PHP
107                LDA 87
108                LSR
109                LSR
110                LSR
111                LSR
112                PLP
113                ADC SC+5
114                STA SC+5
115                CMP #$3A
116                BCC *+5
117                SBC #10
118                STA SC+5
119                TYA
120                LDX #4
121SC2             ADC #0
122                ADC SC,X
```

```
123              STA SC,X
124              CMP #$3A
125              BCC *+5
126              SBC #10
127              STA SC,X
128              LDA #0
129              DEX
130              BNE SC2
131              RTS
132INVADER       LDA BOBZ
133              ORA BBZ
134              BEQ *+1
135RTS           RTS
136              LDA ISF
137              BNE ISCH
138              JSR RND
139              CMP #8
140              BCS RTS
141              LDX INVA+1
142              LDA INVA
143              CLC
144              ADC #176
145              STA $FE
146              BCC *+1
147              INX
148              STX $FF
149              JSR RND
150              AND #$1F
151              CMP #22
152              BCS *-9
153              TAY
154IS1           LDA $FE
155              STA ISCHADR
156              LDA $FF
157              STA ISCHADR+1
158              LDX #5
159IS2           LDA (0),Y
160              CMP #64
161              BCS IS3
162              SEC
163              LDA ISCHADR
164              SBC #44
165              STA ISCHADR
166              BCS *+2
167              DEC ISCHADR+1
168              DEX
169              BNE IS2
170              BEQ IS4
171IS3           AND #3
172              CLC
173              BEQ IS5
174              CMP #3
175              BEQ IS5
176IS4           DEY
177              BPL IS1
178              LDY #21
179              BNE IS1
180IS5           BCC *+3
181              LDA #$40
182              DFB $2C,$A9,4
183              LDX #7
184Z57           STA 6528,X
185              DEX
186              BPL Z57
187              STX ISF
188              TYA
189              CLC
190              ADC #22
191              ADC ISCHADR
192              STA ISCHADR
193              BCC *+2
194              INC ISCHADR+1
195              LDA #32
196              STA ZUISCH
197              LDY #0
198              LDA #176
199              STA (ISCHADR),Y
200              RTS
201ISCH          LDY #0
202              LDA ZUISCH
203              STA (ISCHADR),Y
204              CLC
205              LDA ISCHADR
206              ADC #22
207              STA ISCHADR
208              BCC *+2
209              INC ISCHADR+1
210              LDA (ISCHADR),Y
211              STA ZUISCH
212              CMP #32
213              BNE ISS1
214              LDA #176
215              STA (ISCHADR),Y
216              RTS
217ISS1          CMP #76
218              BCC P3
219              CMP #140
220              BCS *+4
221              CMP #84
222              BCS P3
223              CMP #174
224              BCS P3
225              JSR CHRRADR
226              LDY #7
227X20           LDA ($FE),Y
228              STA 6536,Y
229              AND 6528
230              BNE X21
231              DEY
232              BPL X20
233              LDY #7
234X22           LDA ($FE),Y
235              ORA 6528
236              STA 6536,Y
237              DEY
238              BPL X22.
239              INY
240              LDA #177
241              STA (ISCHADR),Y
242              RTS
243X21           LDY #0 ;GETROFF
244              STY ISF
```

Starship VC-20

```
245        LDA ZUISCH
246        CMP #84
247        BCS X23
248        ADC #48
249        TAY
250        LDA BP
251        LSR
252        LSR
253        TAX
254        TYA
255        STA BLA,X
256        CLC
257        ADC #1
258        STA BLA+1,X
259        LDA #10
260        STA BBZ
261        STA BDZ
262        RTS
263X23     LDA 6528
264        EOR #$FF
265        LDY #7
266        PHA
267X51     PLA
268        PHA
269        AND ($FE),Y
270        STA ($FE),Y
271        DEY
272        BPL X51
273        PLA
274        INY
275        STY ISF
276        RTS
277P3      LDX #20 ;BODEN
278        STX BOBZ
279        STX BDZ
280        LDY #0
281        STY ISF
282        LDA 6528
283        CMP #64
284        BCC BOD1
285        LDX ISCHADR
286        BNE *+2
287        DEC ISCHADR+1
288        DEC ISCHADR
289BOD1     LDA #0
290        ROL
291        ASL
292        ADC #120
293        STA (ISCHADR),Y
294        INY
295        ADC #1
296        STA (ISCHADR),Y
297        RTS
298SSCHINC  LDA SSF
299        BNE *+1
300        RTS
301        LDY #0
302        LDA ZUSSCH
303        STA (SSCHADR),Y
304        SEC
305        LDA SSCHADR
306        SBC #22
307        STA SSCHADR
308        BCS *+2
309        DEC SSCHADR+1
310        CMP #66
311        LDA SSCHADR+1
312        SBC #$10
313        BCS SF2
314        LDX HIM
315        CPX #3
316        BCC *+8
317        LDA SSCHADR
318        BNE *+2
319        DEC SSCHADR+1
320        DEC SSCHADR
321        TXA
322        ASL
323        CLC
324        STY SSF
325        ADC #112
326        STA (SSCHADR),Y
327        INY
328        ADC #1
329        STA (SSCHADR),Y
330        LDX #20
331        STX HIMZ
332        STX SBDZ
333        RTS
334SF2      LDY #0
335        LDA (SSCHADR),Y
336        STA ZUSSCH
337        CMP #32
338        BNE *+5
339        LDA #178
340        STA (SSCHADR),Y
341        RTS
342        JSR CHRRADR
343        LDY #7
344SF3      LDA ($FE),Y
345        STA 6552,Y
346        AND 6544
347        BNE SFG
348        DEY
349        BPL SF3
350        LDY #7
351SF4      LDA ($FE),Y
352        ORA 6544
353        STA 6552,Y
354        DEY
355        BPL SF4
356        INY
357        LDA #179
358        STA (SSCHADR),Y
359        RTS
360SFG      LDA ZUSSCH ;GETROFFEN
361        LDY #0
362        STY SSF
363        CMP #140
364        BCC SF5
365        CMP #174
366        BCS SF5
```

```
367            LDA 6544
368            EOR #$FF
369            LDY #7
370            PHA
371            PLA
372            PHA
373            AND ($FE),Y
374            STA ($FE),Y
375            DEY
376            BPL *-9
377            PLA
378            RTS
379SF5         CMP #76
380            BCS SF6
381            STA BI
382            LSR
383            BCC SF7
384            LDA SSCHADR
385            BNE *+2
386            DEC SSCHADR+1
387            DEC SSCHADR
388SF7         LDX #10
389            STX IBZ
390            INC SBDZ
391            RTS
392SF6         LDX #21
393            LDA #32
394            STA ULA,X
395            DEX
396            BNE *-6
397            STX SSF
398            STX UFF
399            JSR RND
400            CMP #254
401            BCC *+4
402            LDA #$60
403            BNE JSC
404            CMP #252
405            BCC *+4
406            LDA #$50
407            BNE JSC
408            CMP #248
409            BCC *+4
410            LDA #$40
411            BNE JSC
412            CMP #240
413            BCC *+4
414            LDA #$30
415            BNE JSC
416            CMP #224
417            BCC *+4
418            LDA #$20
419            BNE JSC
420            LDA #$15
421JSC         JMP SCO
422TAST        LDA 197
423            CMP #32
424            BEQ SS
425            LDA BP
426            LSR
427            LSR
```

```
428            TAX
429            LDA #32
430            STA BLA,X
431            STA BLA+1,X
432            LDA 197
433            CMP #30
434            BNE *+10
435            LDY BP
436            CPY #83
437            BEQ *+3
438            INC BP
439            CMP #33
440            BNE *+8
441            LDY BP
442            BEQ *+3
443            DEC BP
444            LDA BP
445            LSR
446            LSR
447            TAX
448            LDA BP
449            AND #3
450            ASL
451            ADC #76
452            STA BLA,X
453            ADC #1
454            STA BLA+1,X
455T1         RTS
456SS         LDA SSF
457            ORA IBZ
458            ORA HIMZ
459            BNE T1
460            LDA BP
461            AND #3
462            TAX
463            STX HIM
464            LDA TAB,X
465            LDY #7
466Z34T        STA 6544,Y
467            DEY
468            BPL Z34T
469            LDA BP
470            LSR
471            LSR
472            CPX #3
473            SBC #21
474            ADC #>BLA
475            STA SSCHADR
476            LDA #<BLA
477            SBC #0
478            STA SSCHADR+1
479            STA SSF
480            LDA #32
481            STA ZUSSCH
482            LDY #0
483            LDA #178
484            STA (SSCHADR),Y
485Z449        RTS
486UFO         LDA UFF
487            BNE INCU
488            JSR RND
```

```
489         CMP #254
490         BCC Z449
491         STA UFF
492         LDA #0
493         STA UP
494         LDA #84
495         STA ULA
496         LDA #85
497         STA ULA+1
498         LDA #86
499         STA ULA+2
500         RTS
501INCU     LDA UP
502         LSR
503         LSR
504         LSR
505         TAX
506         LDA #32
507         STA ULA,X
508         STA ULA+1,X
509         STA ULA+2,X
510         INC UP
511         LDA UP
512         EOR #160
513         BNE *+4
514         STA UFF
515         RTS
516         EOR #160
517         LSR
518         LSR
519         LSR
520         TAX
521         LDA UP
522         AND #7
523         STA $FE
524         ASL
525         ADC $FE
526         ADC #84
527         STA ULA,X
528         ADC #1
529         STA ULA+1,X
530         ADC #1
531         STA ULA+2,X
532         RTS
533INVM     DEC MDZ
534         BMI *+1
535         RTS
536         LDA IBZ
537         BNE *-6
538         LDX #4
539         STX MDZ
540         LDX LIZ
541         DEX
542         BPL *+2
543         LDX #4
544         STX LIZ
545         PHP
546         LDA INVA+1
547         STA $FF
548         LDA INVA
549IM1      CLC
```

```
550         DEX
551         BMI IM2
552         ADC #44
553         BCC IM1
554         INC $FF
555         BCS IM1
556IM2      STA $FE
557         LDY #21
558         LDA #0
559         ORA ($FE),Y
560         DEY
561         BPL *-5
562         AND #2
563         CMP #2
564         LDX MR
565         BNE IM3
566         LDA $FE
567         ADC #0
568         STA 88
569         LDA $FF
570         ADC #0
571         STA 89
572IM4      LDY #21
573         LDA ($FE),Y
574         PHA
575         LDA #32
576         STA ($FE),Y
577         PLA
578         CMP #64
579         BCC IM41
580         EOR #2
581         STA (88),Y
582IM41     DEY
583         BPL IM4+2
584         PLP
585         BEQ *+1
586         RTS
587         LDX INVA+1
588         LDA INVA
589         STX $FF
590         STA $FE
591         LDA #0
592         LDY #21
593         LDX #5
594IM5      ORA ($FE),Y
595         PHA
596         LDA $FE
597         CLC
598         ADC #44
599         STA $FE
600         BCC *+2
601         INC $FF
602         PLA
603         DEX
604         BNE IM5
605         AND #3
606         CMP #3
607         BEQ *+1
608         RTS
609         DEC MR
610         JMP MD
```

```
611IM3   LDY #0
612      LDA $FE
613      SBC #0
614      STA 88
615      LDA $FF
616      SBC #0
617      STA 89
618IM6   LDA ($FE),Y
619      PHA
620      LDA #32
621      STA ($FE),Y
622      PLA
623      CMP #64
624      BCC IM61
625      EOR #2
626      STA (88),Y
627IM61  INY
628      CPY #22
629      BNE IM6
630      PLP
631      BEQ *+1
632      RTS
633      LDY #0
634      STY 87
635      LDX INVA+1
636      LDA INVA
637      STX $FF
638      STA $FE
639      LDX #5
640IM7   LDA ($FE),Y
641      CMP #64
642      BCC IM8
643      EOR #2
644      ORA 87
645      STA 87
646IM8   LDA $FE
647      CLC
648      ADC #44
649      STA $FE
650      BCC *+2
651      INC $FF
652      DEX
653      BNE IM7
654      LDA 87
655      AND #2
656      BNE *+1
657      RTS
658      INC MR
659MD    LDA INVA
660      STA 88
661      CLC
662      ADC #22
663      STA 90
664      STA INVA
665      LDA INVA+1
666      STA 89
667      ADC #0
668      STA INVA+1
669      STA 91
670      LDY #198
671MD1   DEY
672      LDA (88),Y
673      STA (90),Y
674      TYA
675      BNE MD1
676      LDA #32
677      LDY #21
678      STA (88),Y
679      DEY
680      BPL *-5
681      LDA 89
682      CLC
683      ADC #132
684      STA 89
685      LDA 91
686      ADC #132
687      STA 91
688      LDY #198
689MD2   DEY
690      LDA (88),Y
691      STA (90),Y
692      TYA
693      BNE MD2
694      LDX #22
695MD3   LDA 4513,X
696      CMP #64
697      BCC MD4
698      CMP #76
699      BCC MD5
700MD4   DEX
701      BNE MD3
702      RTS
703MD5   INC 3
704      RTS
705BBL   LDA BBZ
706      BEQ BB1
707      DEC BDZ
708      BEQ *+1
709BB1   RTS
710      LDA #8
711      STA BDZ
712      DEC BBZ
713      BNE BB2
714      LDX #22
715      LDA #32
716BB3   STA BLA-1,X
717      DEX
718      BNE BB3
719      STX ISF
720      DEC AZB
721      BNE *+2
722      INC 3
723      RTS
724BB2   LDA BP
725      LSR
726      LSR
727      TAX
728      LDA BLA,X
729      CMP #132
730      BCC *+4
731      SBC #8
732      BCS *+2
```

Starship VC-20

```
733                ADC #8
734                STA BLA,X
735                CLC
736                ADC #1
737                STA BLA+1,X
738                RTS
739HIBL            LDA HIMZ
740                BEQ HI1
741                DEC SBDZ
742                BEQ *+1
743HI1             RTS
744                LDA #8
745                STA SBDZ
746                DEC HIMZ
747                BEQ HI2
748                LDA SSCHADR+1
749                CLC
750                ADC #132
751                STA $FF
752                LDA SSCHADR
753                STA $FE
754                LDY #0
755                LDA ($FE),Y
756                EOR #22
757                STA ($FE),Y
758                INY
759                STA ($FE),Y
760                RTS
761HI2             LDY #0
762                LDA #32
763                STA (SSCHADR),Y
764                INY
765                STA (SSCHADR),Y
766                RTS
767BOBL            LDA BOBZ
768                BEQ BO1
769                DEC BDZ
770                BEQ *+1
771BO1             RTS
772                LDA #8
773                STA BDZ
774                DEC BOBZ
775                BEQ BO2
776                LDA ISCHADR+1
777                CLC
778                ADC #132
779                STA $FF
780                LDA ISCHADR
781                STA $FE
782                LDY #0
783                LDA ($FE),Y
784                EOR #22
785                STA ($FE),Y
786                INY
787                STA ($FE),Y
788                RTS
789BO2             LDY #0
790                LDA #32
791                STA (ISCHADR),Y
792                INY
793                STA (ISCHADR),Y
794                RTS
795IBL             LDA IBZ
796                BEQ *+5
797                DEC SBDZ
798                BEQ *+1
799                RTS
800                LDA #5
801                STA SBDZ
802                DEC IBZ
803                BEQ IB1
804                LDY #0
805                LDA (SSCHADR),Y
806                CMP #108
807                LDA BI
808                BCS IB2
809                AND #2
810                ADC #108
811IB3             STA (SSCHADR),Y
812                ADC #1
813                INY
814                STA (SSCHADR),Y
815                RTS
816IB2             AND #$7E
817                CLC
818                BNE IB3
819IB1             LDY #0
820                LDA #32
821                STA (SSCHADR),Y
822                INY
823                STA (SSCHADR),Y
824                LDA BI
825                CMP #68
826                BCS *+4
827                LDA #$10
828                BNE JSC1
829                CMP #72
830                BCS *+4
831                LDA #$6
832                BNE JSC1
833                LDA #2
834JSC1            JSR SCO
835                DEC AZI
836                BEQ *+1
837                RTS
838                LDA #40
839                STA AZI
840                LDA INVI
841                CLC
842                ADC #22
843                STA INVI
844                STA INVA
845                BCC *+6
846                INC INVI+1
847                INC INVA+1
848                INC AZB
849                LDA INVI
850                STA $FE
851                LDA INVI+1
852                CLC
853                ADC #132
854                STA $FF
```

```
855    LDA #F1
856    LDY #43
857    STA ($FE),Y
858    DEY
859    BPL *-5
860    LDA $FE
861    CLC
862    ADC #44
863    STA $FE
864    BCC *+2
865    INC $FF
866    LDA #F2
867    LDY #87
868    STA ($FE),Y
869    DEY
870    BPL *-5
871    LDA $FE
872    CLC
873    ADC #88
874    STA $FE
875    BCC *+2
876    INC $FF
877    LDY #220
878    LDA #F3
879    DEY
880    STA ($FE),Y
881    TYA
882    BNE *-8
883    LDA #0
884    STA LIZ
885    STA MR
886    LDA INVA
887    STA $FE
888    LDA INVA+1
889    STA $FF
890    LDA #64
891    LDY #14
892    STA ($FE),Y
893    DEY
894    DEY
895    BPL *-6
896    LDA #65
897    LDY #15
898    STA ($FE),Y
899    DEY
900    DEY
901    BPL *-6
902    LDA $FE
903    CLC
904    ADC #44
905    STA $FE
906    BCC *+2
907    INC $FF
908    LDA #68
909    LDY #14
910    STA ($FE),Y
911    DEY
912    DEY
913    BPL *-6
914    LDA #69
915    LDY #15
916    STA ($FE),Y
917    DEY
918    DEY
919    BPL *-6
920    LDA $FE
921    CLC
922    ADC #44
923    STA $FE
924    BCC *+2
925    INC $FF
926    LDA #68
927    LDY #14
928    STA ($FE),Y
929    DEY
930    DEY
931    BPL *-6
932    LDA #69
933    LDY #15
934    STA ($FE),Y
935    DEY
936    DEY
937    BPL *-6
938    LDA $FE
939    CLC
940    ADC #44
941    STA $FE
942    BCC *+2
943    INC $FF
944    LDA #72
945    LDY #14
946    STA ($FE),Y
947    DEY
948    DEY
949    BPL *-6
950    LDA #73
951    LDY #15
952    STA ($FE),Y
953    DEY
954    DEY
955    BPL *-6
956    LDA $FE
957    CLC
958    ADC #44
959    STA $FE
960    BCC *+2
961    INC $FF
962    LDA #72
963    LDY #14
964    STA ($FE),Y
965    DEY
966    DEY
967    BPL *-6
968    LDA #73
969    LDY #15
970    STA ($FE),Y
971    DEY
972    DEY
973    BPL *-6
974    RTS
```

Editor für 6502-Systeme

Seine Reihe „Programmieren in Maschinensprache“ setzt Homecomputer diesesmal fort mit dem Editor.

Sie haben nun eine Programmserie, mit der Sie Programme und Unterprogramme in der Assemblersprache für alle Microcomputer mit dem 6502-Prozessor schreiben können.

Obwohl die Reihe speziell für den VC-20 und VC-64 begonnen wurde, sind die Programme fast ohne Umschreiben auch für die CBMs, den Apple, den Atari, usw. verwendbar. Lediglich die Bildschirm-Befehle und Tastatur-POKEs müssen angepaßt werden bzw. können einfach ignoriert werden.

Der Assembler, der in der April-Ausgabe abgedruckt wurde, hat früher als vorgesehen schon weitere Zusatzfunktionen erhalten, denen in den nächsten Monaten von Zeit zu Zeit weitere folgen werden.

Auch der Disassembler aus Heft 3, wo übrigens für den VC-20 zwei Zeilen beim Druck verlorengegangen sind, wurde entsprechend angepaßt.
Man beginnt mit dem Einladen des Editors. Nach dem Starten erhält man eine Tabelle mit den Befehlen, die der Editor kennt und den möglichen Parametern.

Es genügt, wenn die inversen Buchstaben eingegeben werden und auch Blanks sind optional. Man kann aber auch den gesamten Befehl eingeben oder Teile davon, sofern die signifikanten Buchstaben am Anfang enthalten sind.

```
50 PRINT"██HOMECOMPUTER-ASSEMBLER█
100 READO$:READA$:O$=O$+A$:READA$:O$=O$+A$:DIMA$(500),A(500)
110 FORI=0TO7:READAI$(I):NEXT
120 DIMO%(58):FORI=0TO55:READO%(I):NEXT
130 FORI=0TO7:READS%(I):NEXT
140 DIMAD%(21):FORI=0TO21:READAD%(I):NEXT
200 PRINT"CASSETTE / DISK █C█/█D█":WAIT198,1:GETG$:IFG$<"C"ORG$>"D"THEN200
210 DI=G$="D":IFNOTDITHENGOSUB50000
220 INPUT"FILENAME";F$:IFDITHENF$=F$+",S,R"
500 OPEN1,1+(7ANDDI),2ANDDI,F$:Z=0
510 Z=Z+1:INPUT#1,L$:FORI=1TO9:IFMID$(L$,I,1)<>" "THENNEXT
520 IFI-1THENA$(LC)=LEFT$(L$,I-1):A(LC)=PC:LC=LC+1
525 PRINTRIGHT$("     "+STR$(PC),5)LEFT$(L$,I)TAB(10)MID$(L$,I+1)
530 GOSUB4000
540 PC=PC+L
900 IF(64ANDST)=0THEN510
910 CLOSE1
1000 IFNOTDITHENGOSUB50000
1010 OPEN1,1+(7ANDDI),2ANDDI,F$:F=-1:Z=0
1020 Z=Z+1:INPUT#1,L$:FORI=1TO9:IFMID$(L$,I,1)<>" "THENNEXT
1030 GOSUB4000:IFOI>55THEN1150
1040 O=O%(OI):IFOI>21THEN1070
1050 IFAI=1ANDOAND1ORL=3THENO=O+8
1060 O=O+S%(AI)
1070 IFL>1THENGOSUB5000:IFOI>46ANDOI<55THENS=S-PC-2:S=S+(256ANDS<0)
1080 S%=S/256:S=S-256*S%:PRINTPCTAB(7)OTAB(12)STAB(17)S%
1090 POKEPC,O:POKEPC+1,S:IFL=3THENPOKEPC+2,S%
1100 PC=PC+L:IF(64ANDST)=0THEN1020
1110 CLOSE1
1120 GOTO20000
1150 IFOI-57THEN1100
1160 K=0:FORP=PCTOP+L-1:A$="":FORK=K+1TOLEN(L$):J$=MID$(L$,K,1)
1165 IFJ$<>","THENA$=A$+J$:NEXT
1170 GOSUB5000:POKEP,S:NEXTP:GOTO1100
4000 A$=MID$(L$,I+1,3):L=0
4010 FORJ=1TO175STEP3:IFA$<>MID$(O$,J,3)THENNEXT:PRINT"OPCODE ERROR IN"Z:GOTO200
00
4020 OI=(J-1)/3:L$=MID$(L$,I+5)
4030 IFOI=56ANDNOTFTHENA$=L$:GOSUB5000:A(LC-1)=S
4032 IFOI=57THENL=1:FORJ=1TOLEN(L$):L=L-(MID$(L$,J,1)=","):NEXT
4035 IFOI=58THENA$=L$:GOSUB5000:PC=S
4040 IFOI>55THENRETURN
4045 IFOI>21THENA$=L$:GOTO4135
4050 IFLEN(L$)=0ORL$="A"THENB$="A":GOTO4140
4060 A$="":B$="":FORI=1TOLEN(L$):J$=MID$(L$,I,1)
4070 IFJ$="#"ORJ$="("THENB$=B$+J$:GOTO4130
4090 IFJ$=")"ORJ$=" "THEN4130
4100 IFJ$=","THENI=I+1:B$=B$+MID$(L$,I,1):GOTO4130
4110 IFJ$=";"THEN4135
4120 A$=A$+J$
4130 NEXT
4135 L=-(OI>21)-(OI>46)-(OI>54):IFLTHENRETURN
4140 FORAI=0TO7:IFB$<>AI$(AI)THENNEXT:GOTO19990
4150 L=1-(AI>0)-(AI>3):IF(OI=12ORI=15)ANDAI=4THENAI=7
4152 IF(2↑AIANDAD%(OI))=0THEN19990
4155 IFAI<6THENRETURN
4160 GOSUB5000:IFE=1ORI=7THENL=3:RETURN
4170 L=2-(S>255)
4175 IFAI=7THENIFL=3AND(OI=10ORI=12)THEN19990
4180 RETURN
```

Starship VC-20

```
•5000 FORI=1TOLEN(A$):J=ASC(MID$(A$,I,1))
•5010 IFJ=43THENSA$=MID$(A$,I+1):A$=LEFT$(A$,I-1):GOSUB5040:SA=S:A$=SA$:GOSUB5040
 :S=S+SA:RETURN
•5020 IFJ=45THENSA$=MID$(A$,I+1):A$=LEFT$(A$,I-1):GOSUB5040:SA=S:A$=SA$:GOSUB5040
 :S=SA-S:RETURN
•5030 NEXT
•5040 IFASC(A$)=62THENA$=MID$(A$,2):GOSUB6000:S%=S/256:S=S-256*S%:RETURN
•5050 IFASC(A$)=60THENA$=MID$(A$,2):GOSUB6000:S=INT(S/256):RETURN
•6000 E=0:S=VAL(A$):IFS>0ORASC(A$)=48THENRETURN
•6010 J=ASC(A$):IFJ-36THEN6030
•6020 FORJ=2TOLEN(A$):J$=MID$(A$,J,1):S=16*S+VAL(J$)+(ASC(J$)-55ANDJ$>"@"):NEXT:R
ETURN
•6030 IFJ=37THENFORJ=2TOLEN(A$):S=S+S+VAL(MID$(A$,J,1)):NEXT:RETURN
•6035 IFJ=42THENS=PC+L:RETURN
•6040 FORJ=0TOLC-1:IFA$=A$(J)THENS=A(J):RETURN
•6050 NEXT:E=1:RETURN
 10000 DATAORAASLBITANDROLEORLSRJMPADCRORSTYSTASTXLDYLDALDXCPYCMPDECCPXSBCINC
 10010 DATABRKPHPCLCPLPSECRTIPHACLIRTSPLASEIDEYTXATYATXSTAYTAXCLVTSXINYDEXCLDINXN
 OP
 10020 DATASEDBPLBMIBVCBVSBCCBCSBNEBEQJSREQUDFBORG
 10030 DATAA,#,(X,(Y,Y,(,"",X
 10040 DATA5,6,36,37,38,69,70,68,101,102,132,133,134,164,165,166,196,197,198,228,
 229
 10050 DATA230,0,8,24,40,56,64,72,88,96,104,120,136,138,152,154,168,170,184,186,2
 00
 10060 DATA202,216,232,234,248,16,48,80,112,144,170,208,240,32
 10070 DATA4,-4,-4,12,12,32,0,16
•10080 DATA222,193,64,222,193,222,193,96,222,193,192,222,194,194,222,194,66,222,2
 04,66
•10090 DATA222,204
•19990 PRINT"ADR ERROR IN"Z
•20000 PRINT"EINLADEN VOM:
•20005 PRINT"EDITOR ?":POKE198,0:WAIT198,1:GETG$:IFG$="J"THENCLR:LOAD"EDITOR",8
•20010 PRINT"DISASSEMBLER ?":WAIT198,1:GETG$:IFG$="J"THENCLR:LOAD"DISASSEMBLER",
 8
•20020 END
 50000 PRINT"CASSETTE ZURUECKSPULEN":PRINT"DANN  P  DRUECKEN!"
 50005 GETWA$:IFWA$<>"P"THEN50005
 50010 PRINT:RETURN
•60000 INPUTA$:A$="%"+A$:GOSUB6000:PRINTS:GOTO60000

READY.
```

Die Zeilennummern mit Punkt wurden gegenüber der Heft 4/83-Version geändert. Für Kassettenbetrieb entfallen in den Zeilen 20005 und 20010 die letzten beiden Zeichen.

ADD
Um ein Assembler-Programm einzugeben, benutzt man den Befehl A(DD) (Anm.: die Buchstaben in Klammern sind optional).
Dann erscheint eine “1“ mit einem inversen Blank dahinter. Dies ist der Cursor, während die “1“ die laufende Zeilennummer des Befehls ist. Wenn man einmal einen Befehl unter einer bestimmten Zeilennummer eingegeben hat, so heißt dies nicht, daß zu dieser Zeilennummer immer der eingegebene Befehl gehört, denn man kann auch noch nachträglich Befehle einfügen, und dabei ändern sich dann entsprechend die Nummern der folgenden Befehle. ADD heißt immer Anhängen von weiteren Befehlen. Wenn der Speicher leer ist, dann lautet die „nächste“ Zeile 1, wenn bereits 43 Zeilen vorhanden sind, dann ist 44 die Zeile, in die nach ADD geschrieben wird.

INSERT
Hat man irgendwo im Assembler-Listing einen oder mehrere Befehle vergessen und nachzutragen, so benutzt man den Befehl I(NSERT). Und zwar gibt man nach dem Befehlswort die Zeilennummer an, hinter der die Befehle eingeschoben werden sollen. Fehlen Befehle direkt am Anfang, so wird nur der Befehl ohne Parameter eingegeben und die Zeilen werden solange eingeschoben, bis eine neue Editoranweisung dieses beendet.

DELETE
Sind Assembler-Zeilen überflüssig geworden, so kann man sie mit D(ELETE) entfernen. Wie bei dem BASIC-Befehl LIST kann man hinter dem Delete-Befehl entweder nur eine Zeilennummer eingeben, wenn man nur eine Zeile löschen will; oder z. B. “D-5“ eingeben, wenn man die ersten 5 Zeilen löschen möchte; oder “D5-“, wenn alle Zeilen ab Nummer 5 gelöscht werden sollen; oder einfach “D“, um sämtliche Zeilen zu löschen.
Damit man nicht aus Versehen Zeilen entfernt, die man noch benötigt, wird zur Sicherheit noch einmal gefragt, ob man auch sicher ist. Die Frage (in Englisch) taucht immer dann auf, wenn mehr als 10 Zeilen gelöscht werden sollen.

LIST
Auch bei L(IST) ist die Parameterübergabe wie im BASIC. Wird beim Listen die Taste “S“ gedrückt, so stoppt der Editor und erwartet zur Fortsetzung einen weiteren Tastendruck. Drücken der Taste “Q“ beendet das Listen bevor die letzte Zeilennummer erreicht ist.

EDIT
Stellt man fest, daß eine Zeile falsch eingegeben wurde, so kann sie mit dem E(DIT)-Befehl korrigiert werden. Die Parameterübergabe ist wie unter DELETE beschrieben. Der Editor listet die zu korrigierende Zeile und setzt den Cursor an den Zeilenanfang.
Diesmal ist es ein „echter Cursor“ und es zählt, wie vom CBM gewohnt, alles, was in der oder den entsprechenden Bildschirmzeilen steht. Deshalb kann man nach Durchführen der Korrektur sofort “RETURN“ drücken, und auch der Rest nach dem Cursor wird übernommen. Oder nur “RETURN“, wenn in der Zeile nichts geändert werden soll.
Auch bei dem EDIT-Befehl kann man einfach eine neue Editor-Anweisung eingeben, wenn man aufhören will, sofern man den Rest der Zeile (oder Zeilen) mit Spaces löscht.

COPY
Wenn man einen Block, den man schon einmal eingegeben hat, an einer anderen Stelle haben möchte, so verwendet man den Befehl C(OPY). Direkt hinter dem Befehl gibt man aber noch keine Parameter an. Der Editor fragt diese näm-

```
80 POKE45,32:POKE46,31
90 CLR
100 DIMA$(500)
110 L$=" ":FORI=1TO5:L$=LEFT$(L$+L$,21):NEXT:L$=L$+"
120 PRINTTAB(6)"HOMECOMPUTER
200 PRINT"HOMECOMPUTER EDITOR
210 PRINT"APPEND [A]
215 PRINT"SAVE [A][-][E]
220 PRINT"VERIFY [A][-][E]
225 PRINT"ADD
230 PRINT"INSERT [A]
235 PRINT"EDIT [A][-][E]
240 PRINT"DELETE [A][-][E]
245 PRINT"LIST [A][-][E]
250 PRINT"COPY
255 PRINT"QUIT -> ASS + DISASS
280 IFB=1ORB=5ORB=9THENA$="L":GOTO300
290 INPUT"BEFEHL";A$
300 PRINT:RESTORE:FORB=1TO10:READB$:IFLEFT$(A$,LEN(B$))<>
B$THENNEXT
310 IFB<>2ANDB<9THENGOSUB5000
330 ONBGOTO400,600,500,700,800,1000,1200,900,1100,540,200
400 REM APPEND
410 GOSUB6000
420 INPUT#1,A$
430 IFN2<NTHENFORC=NTON2+1STEP-1:A$(C+1)=A$(C):NEXT
440 N2=N2+1:N=N+1:A$(N2)=A$:IF(64ANDST)=0THEN420
450 CLOSE1:GOTO280
500 REM SAVE
502 INPUT"FILENAME";F$:PRINT"CASSETTE / DISK C/D":WAI
T198,1:GETG$:IFG$="C"THEN510
505 IFG$<>"D"THEN502
507 OPEN1,8,2,F$+",S,W":GOTO520
510 OPEN1,1,1,F$
520 FORC=N1TON2:PRINT#1,CHR$(34)A$(C):NEXT:CLOSE1
540 PRINT"ASSEMBLER EINLADEN ?
550 WAIT198,3:GETG$
560 IFG$="J"THENCLR:LOAD"ASSEMBLER",8
570 PRINT"DISASSEMBLER ?       "
580 WAIT198,3:GETG$
590 IFG$="J"THENCLR:LOAD"DISASSEMBLER",8
595 GOTO200
600 REM ADD
610 C=N:PRINT"
620 GOSUB4000:IFBFTHEN300
630 PRINTL$:GOSUB4500
640 A$(C)=A$:N=C:GOTO620
700 REM LIST
710 PRINT"":FORC=N1TON2:A$=A$(C):GOSUB4500
720 GETG$:IFG$="S"THENWAIT198,1:GETG$
730 IFG$="Q"THEN280
740 NEXT:GOTO280
800 REM DELETE
810 D=N2-N1+1:IFD>10THENPRINT" ARE YOU SURE ? ":WAIT198
,7:GETG$:IFG$="N"THEN280
820 N=N-D
830 FORC=N1TON:A$(C)=A$(C+D):NEXT:GOTO280
900 REM VERIFY
910 GOSUB6000
920 FORC=N1TON2:INPUT#1,A$
930 IFA$<>A$(C)OR64ANDSTANDC<N2THENPRINT"?VERIFY ERROR IN
"C:CLOSE1:GOTO280
940 NEXT:CLOSE1:PRINT"OK":GOTO280
```

lich der Reihe nach ab. Er beginnt mit dem Bereich der zu kopierenden Zeilen. Die Zeilennummern werden wie bei DELETE, LIST oder EDIT behandelt.
Danach erfolgt nach Anweisung die Eingabe der Zeilennummer, hinter der die Zeilen eingeschoben werden sollen (fast wie bei INSERT).
Man kann den Block nämlich nicht nur anhängen, sondern an jede Stelle innerhalb des Assembler-Listings einfügen, sogar innerhalb des Blocks selbst. Lediglich wenn man den Block direkt an den Anfang kopieren möchte muß man sich eines kleinen Tricks behelfen. Man fügt dann nämlich vorher einfach eine neue Zeile 1 ein, kopiert den Block (Vorsicht! Die zu kopierenden Zeilennummern haben sich jetzt um 1 verschoben!) hinter die Zeile 1 und löscht dieselbe mit D1. Der zu kopierende Block steht jetzt am Anfang.

APPEND SAVE VERIFY

Drei Befehle dienen dem Laden, Retten und Verifizieren von Assembler-Files auf Diskette oder Kassette.
Der Befehl LOAD, den man in diesem Zusammenhang erwartet gibt es nicht. Ihn ersetzt der Befehl AP(PEND), der einen Parameter haben kann; die Zeilennummer, hinter der das File vom Datenträger in das im RAM befindliche Programm geladen werden soll.
Ist der Speicher leer, so wird sich nach APPEND nur das Programm, das zugeladen wurde, im Speicher befinden.
APPEND ohne Parameter hängt das Programm an das im Speicher befindliche an!
Um das Assembler-Listing auf Kassette oder Disk auf-

zunehmen, wird der Befehl S(AVE) benutzt. Ohne Parameter speichert man das gesamte, im Speicher befindliche Programm; und mit entsprechenden Parametern (wie oben unter DELETE beschrieben) können Teile des Programms übertragen werden.
Um sich zu vergewissern, daß alles richtig gespeichert wurde, kann mit V(ERIFY) die Korrektheit der Aufnahme überprüft werden. Dabei ist zu beachten, daß wenn Parameter übergeben wurden, diese identisch sind mit denen, die beim Kommando SAVE verwendet wurden.

GOTO 280
Hat man aus irgend einem Grunde Stop oder Break gedrückt, so kann mit GOTO 280 der Programmablauf fortgesetzt werden, ohne daß das Assembler-Listing gelöscht wird (was ja nach RUN der Fall wäre).

QUIT
QU(IT) sorgt als Befehlseingabe dafür, daß Assembler oder Disassembler ausgeführt werden.
Achtung! Zuerst muß das Assembler-Programm im Speicher auf einen Datenträger (Kassette oder Diskette) aufgenommen werden! Erst dann kann das Programm assembliert werden.
Ist der Assembler nachgeladen, dann werden wie vom Bildschirm her befohlen, die Kassette zurückgespult und die weiteren Befehle befolgt. Dies entfällt bei Diskettenbetrieb, wo lediglich darauf zu achten ist, daß sich Assembler und Assembler-Programm auf der selben Diskette befinden.

```
1000 REM EDIT
1010 FORC=N1TON2:PRINT"▓▓TTT":S=PEEK(214):A$=A$(C):GOSUB4500
1015 PRINT"▒";:IFPEEK(214)>STHEN1015
1020 A$="":SYS50528
1030 FORI=515TO600:IFPEEK(I)THENA$=A$+CHR$(PEEK(I)):NEXT
1040 GOSUB4100:IFBFTHEN300
1050 A$(C)=A$:NEXTC:GOTO280
1100 REM COPY
1110 INPUT"COPY-BEREICH";A$
1120 GOSUB5000:INPUT"WOHIN (ZEILENNUMMER)";C
1130 IFC>NTHENC=N
1140 D=N2-N1+1
1150 IFC<NTHENFORI=NTOC+1STEP-1:A$(I+D)=A$(I):NEXT
1160 N=N+D:IFN1>CTHENN1=N1+D
1170 FORI=C+1TOC+D:A$(I)=A$(N1):IFN1=CTHENN1=N1+D
1180 N1=N1+1:NEXT
1190 GOTO280
1200 REM INSERT
1210 C=N1
1220 GOSUB4000:IFBFTHEN300
1230 PRINTL$:GOSUB4500
1240 FORI=NTOCSTEP-1:A$(I+1)=A$(I):NEXT:N=N+1:A$(C)=A$
1250 GOTO1220
4000 A$="":I=1:C=C+I:C$=RIGHT$("  "+STR$(C),3)
4005 PRINT"▓▓TTT
4010 S=PEEK(214):PRINTC$LEFT$(A$,I-1)"▓"LEFT$(MID$(A$,I,1)+" ",1)"■
"MID$(A$,I+1)"  ▒
4015 IFPEEK(214)>STHENPRINT"▒";:GOTO4015
4020 WAIT198,15:GETG$:J=ASC(G$)
4030 IFJ=29THEN4090
4040 IFJ=157THENI=I+(I>1):GOTO4010
4050 IFJ=148THENA$=LEFT$(A$,I-1)+" "+MID$(A$,I):GOTO4010
4060 IFJ=20THENA$=LEFT$(A$,I-1+(I>1))+MID$(A$,I):I=I+(I>1):GOTO4010
4070 IFJ=13THEN4100
4080 A$=LEFT$(A$,I-1)+G$+MID$(A$,I+1)
4090 I=I+1+(I>LEN(A$)):GOTO4010
4100 FORI=1TOLEN(A$):IFMID$(A$,I,1)<>" "THENNEXT:BF=-1:RETURN
4110 IFMID$(A$,I+1,1)=" "THENA$=LEFT$(A$,I)+MID$(A$,I+2):GOTO4110
4120 BF=0:RETURN
4500 PRINTRIGHT$("  "+STR$(C),3);
4510 FORI=1TO8:I$=MID$(A$,I,1):PRINTI$;:IFI$<>" "THENNEXT:I=I-1
4520 PRINTTAB(10)MID$(A$,I+1)"  ":RETURN
5000 D=0:FORI=1TO9:J=ASC(MID$(A$,I,1)+"▓"):IFJ=45THEND=1
5010 IFJ<48ORJ>57THENNEXT
5020 N1=VAL(MID$(A$,I))
5030 N2=VAL(MID$(A$,I+LEN(STR$(N1))))
5040 IFN1=0THENN1=1ANDB<>7:N2=N
5050 IFDTHENN2=N1:N1=1
5060 IFRIGHT$(A$,1)="-"THENN2=N
5070 IFN2<N1THENN2=N1
5080 IFN2>NTHENN2=N
5090 IFN1>NTHENN1=N
5100 RETURN
6000 INPUT"FILENAME";F$:PRINT"CASSETTE / DISK ▓C■/▓D■":WAIT198,1:GE
TG$:IFG$="D"THEN6100
6002 IFG$<>"C"THEN6000
6005 PRINT"CASSETTE ZURUECKSPULEN
6010 PRINT"BIS ZUM ANFANG DES":PRINT"ASSEMBLER-FILES!
6020 PRINT"▓DANN ▓ P ■ DRUECKEN!
6030 WAIT198,1:GETG$:PRINT"OK":OPEN1,1,0,F$:RETURN
6100 OPEN1,8,2,F$+",S,R":RETURN
10000 DATAAP,A,S,L,D,E,I,V,C,Q

READY.
```

Für Kassettenbetrieb entfallen in den Zeilen 560 und 590 die letzten beiden Zeichen.

SYNTAX und Features
Gibt man direkt hinter einer Zeilennummer einen Editorbefehl ein, so wird dieser ausgeführt – und nicht in die Zeile übernommen!
Alles andere, was direkt hinter einer Zeilennummer steht und nach einem Blank mit weiteren Eingaben versehen ist, wird in die entsprechende Zeile geschrieben.
Labels werden direkt hinter die Zeilennummer geschrieben und dürfen keine Blanks enthalten. Ein Blank, das einem Label folgt, bedeutet dem Editor, daß nun ein Mnemonic-Befehl folgt. (Mnemonics, soviel sei für Anfänger gesagt, nennt man die Codewörter wie z. B. LDA (Load Akku), STA (Store Akku), TAX (Tranfer Akku to X-Register), usw. – unser 6502-Lehrgang, der in Kürze beginnt und zu dessen Zweck die Programme vorgestellt wurden, wird darüber Auskunft geben)
Folgt direkt hinter der Zeilennummer ein Blank dann gehört hinter dieses Blank ein Mnemonic.
Dahinter gehört, wenn es sich um einen Mehrbyte-Befehl handelt, wieder ein Blank und dann ein Operand.
Ein Semikolon wird Kommentaren, die zur besseren Verständlichkeit des Programms einer Programmzeile beigefügt werden können, vorangestellt.
Der Assembler versteht Hex- und Dezimalzahlen. Hexzahlen haben ein Dollarzeichen vorangestellt.
Direkte und relative Sprünge zu Labels und zu labelabhängigen Punkten sind möglich. Z. B.:

```
JSR   LABEL
oder
JSR   LABEL+86
BNE   MARKES8-9
```

Auch Sprünge relativ zum Programm-Counter sind gestattet. Das Sternchen "*" ist in diesem Fall das Symbol für den Programm-Counter. Beispiel: BNE. -3
Die bisher zusätzlich vom Assembler verstandenen Mnemonics sind:
ORG Programmbeginn
EQU (Equate)
DFB (Define Byte)
ORG hat kein Label und muß eine Hex- oder Dezimalzahl im Operandenfeld haben.
Vor EQU muß sich ein Label befinden und dahinter eine entsprechende Hex-, Dual- oder Dezimalzahl.
Vor DFB kann ein Label stehen. Dem Mnemonic folgen ein oder mehrere Hex-Dual- oder Dezimalzahlen. Mehrere Zahlen werden durch Kommata getrennt.
Eine weitere nachträglich eingeführte Bequemlichkeit ist, daß mit den Größer-als-und Kleiner-als-Zeichen Labels in High- und Low-Byte zerlegt werden können.
Z. B.

```
1ADRESSE EQU 34356
2        LDA #<ADRESSE
3        STA  HIGHBYTE
4        LDA #>ADRESSE
5        STA  LOWBYTE
```

Wie bereits angesprochen, werden weitere „Bequemlichkeiten“, nützliche Befehle und Hilfsprogramme zum Programmieren in Assembler (und somit in Maschinencode) folgen.

Änderungen gegenüber Disassembler Programm aus Heft 3/83

```
600 INPUT"▒";A$:IFA$="E"THENCLR:LOAD"EDITOR",8
605 IFA$="L"THEN1000
610 IFRIGHT$(A$,1)="L"THENGOSUB800:GOTO1000
690 GOTO600
800 IFLEFT$(A$,1)="$"THEN810
805 PC=VAL(A$):RETURN
810 PC=0:FORI=2TOLEN(A$)-1
820 A=ASC(MID$(A$,I,1))-48:IFA>9THENA=A-7
830 PC=PC*16+A:NEXT:RETURN
900 FORJ=3TO0STEP-1:A=INT(DZ/16↑J):DZ=DZ-A*16↑J:A=A+48:IFA>57THENA=A+7
910 PRINTCHR$(A);:NEXT:RETURN
950 FORJ=1TO0STEP-1:A=INT(DZ/16↑J):DZ=DZ-A*16↑J:A=A+48:IFA>57THENA=A+7
960 PRINTCHR$(A);:NEXT:RETURN
1000 FORI=1TO11:DZ=PC:PRINT"▒";:GOSUB900:PRINT"-▒":P1=PEEK(PC)
1005 ONL%(P1)GOSUB1100,1200,1300
1010 NEXT:GOTO600
1100 DZ=P1:GOSUB950:PRINT"▒";TAB(10);M$(P1):PC=PC+1:RETURN
1200 DZ=P1:GOSUB950:PRINT" ";:P2=PEEK(PC+1):DZ=P2:GOSUB950
1205 PRINT"▒";TAB(10);M$(P1);
1210 IFF%(P1)=9THENP2=PEEK(PC+1):IFP2<128THENP2=PC+P2+2:GOTO1220
1215 IFF%(P1)=9THENP2=PC-(255-P2)+1
1220 PRINT" ";F1$(F%(P1));:DZ=P2:IFF%(P1)=9THENGOSUB900:GOTO1240
1230 GOSUB950
1240 PRINTF2$(F%(P1)):PC=PC+2:RETURN
1300 DZ=P1:GOSUB950:PRINT" ";:P2=PEEK(PC+1):DZ=P2:GOSUB950:PRINT" ";
1310 P3=PEEK(PC+2):DZ=P3:GOSUB950:PRINT"▒";TAB(10);M$(P1);
1320 PRINT" ";F1$(F%(P1));:DZ=P3*256+P2:GOSUB900:PRINTF2$(F%(P1)):PC=PC+3:RETURN

READY.
```

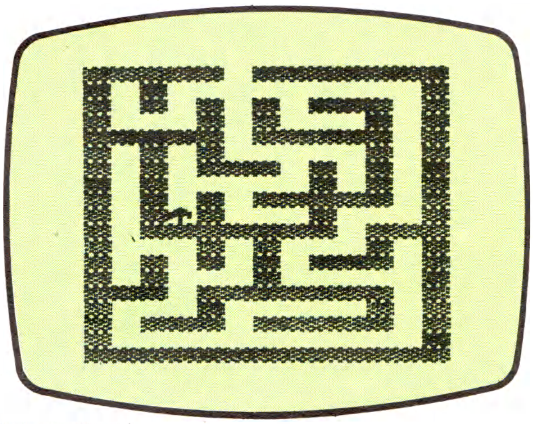
Zeile 602 entfällt
Zeile 1020 entfällt
Für Kassettenbetrieb entfallen in Zeile 600 die letzten zwei Zeichen.

Starship VC-20

Labyrinth in 3,5 K!

Wieviele Leser von Homecomputer uns angerufen haben, haben wir nicht gezählt, aber wenn es um das Labyrinth für den VC-20 ging, dann lag die Antwort auf der Hand, keine Speichererweiterung vorhanden, Programm also nicht lauffähig. Wir haben uns deshalb bemüht (vielen Anrufern habe ich es versprochen), falls es überhaupt möglich ist, das Programm für den VC-20 ohne Erweiterung umzuschreiben, es zum Laufen zu bringen.
Am Ende konnte durch Renummerierung, Dimensionierung, Streichen der Texte und Poken eines neuen RAMTOPs das Programm so weit gekürzt werden, daß es inclusive Variablen fast genau soviel Platz verbraucht, wie ohne Speichererweiterung zur Verfügung stehen.

```
1 DIMM(7),W(3)
2 POKE52,29:POKE54,29:POKE56,29
3 VP=30720:CO=5
4 GOTO79
5 Y=PEEK(TC-U):Z=PEEK(TC-U+LL):IFABS(U)=1ORDC=0THEN7
6 IFY=32ANDZ=32THENFORA=TC-U+LLTOBC-V-LLSTEPLL:POKEA,W3:NEXT
7 POKETC,TW:POKEBC,BW:POKETC+VP,CO:POKEBC+VP,CO
8 B=B+1:IFB<DWTHENTC=TC+U:BC=BC+V:GOTO7
9 IFABS(U)=1THENPOKETC,T1:POKEBC,B1:POKETC+VP,CO:POKEBC+VP,CO
10 IFTC=BC-LLTHENRETURN
11 FORB=TC+LLTOBC-LLSTEPLL:POKEB,SW:POKEB+VP,5:NEXT:RETURN
12 F=ML:U=LL+1:V=1-LL:T1=80:B1=122:SW=103
13 FORX=4TO6STEP2:TW=77:BW=78:B=0:IFX=4THEN15
14 F=MR:U=LL-1:V=-LL-1:SW=101:T1=79:B1=76:TW=78:BW=77
15 LM=M(F):LD=W(F)
16 IFF=3ANDOP=ECTHEN22
17 IFLD=0THENLD=2
18 IFQ-LD*INT(Q/LD)=0THENTC=M(X):BC=M(X+1):GOSUB5:GOTO21
19 TW=99:BW=100:U=U-LL:V=V+LL:N=PEEK(OP+LM)
20 IFN-D*INT(N/D)=0THENTC=M(X)+DW*LL:BC=M(X+1)+DW*-LL:GOSUB5
21 IFABS(U)=1THENU=U+LL:V=V-LL
22 M(X)=M(X)+DW*U:M(X+1)=M(X+1)+DW*V:NEXT
23 IFM=3ANDOP=ECTHENRETURN
24 IFQ-D*INT(Q/D)=0THEN27
25 OP=OP+M(M):Q=PEEK(OP):DC=DC+1:IFDC<CDTHEN12
26 IFDC=CDTHENRETURN
27 R=PEEK(M(4)-1)
28 IFR=32THENM(4)=M(4)-1:M(5)=M(5)-1:IFM(4)<>TL+DW*LLTHEN27
29 R=PEEK(M(6)+1)
30 IFR=32THENM(6)=M(6)+1:M(7)=M(7)+1:IFM(6)<>TR+DW*LLTHEN29
31 FORTC=M(4)TOM(6):POKETC,W1:POKETC+VP,CO:NEXT
32 FORBC=M(5)TOM(7):POKEBC,W2:POKEBC+VP,CO:NEXT:RETURN
33 PRINT"O="OX"LE";
34 GETA$:IFA$=""THEN34
35 PRINT
36 NP=P:Q=PEEK(P):IFA$="G"THENNP=P+M(M)
37 IFNP=PTHEN42
38 IFNP=ECTHENPRINT"ENDE!":GOSUB92:RUN
39 IFQ-D*INT(Q/D)><0THENP=NP:PRINT"SIE GEHEN":GOTO48
40 IFRND(1)>.9THENPRINT"RA";:OL=OL+1
41 PRINT"WAND GERAMMT":GOSUB91:GOTO51
```



```
42 NM=M:IFA$="R"THENNM=M+1
43 IFA$="M"THENGOSUB67
44 IFA$="L"THENNM=M+3
45 IFNM=MTHEN33
46 IFNM>3THENNM=NM-4*INT(NM/4)
47 M=NM:PRINT"DREHEN"
48 D=W(M):Q=PEEK(P):OP=P:ML=M-1:MR=M+1
49 IFML<0THENML=3
50 IFMR>3THENMR=0
51 OX=OX-OL:IFOX<0THENPRINT"LUFT=0":GOSUB92:RUN
52 T=T+1:IFT=TTTHENTT=1:PRINT"LICHT":GOSUB91
53 IFTT=1THENDC=0:GOSUB90:GOSUB12
54 IFP=MCANDRM=0THENRM=1:GOSUB67
55 M(4)=TL:M(5)=BL:M(6)=TR:M(7)=BR:GOTO33
56 P=P+1:IFP>EMTHENP=SM
57 A=INT(RND(1)*4):DC=0
58 A=A+1:DC=DC+1:IFDC>3THEN56
59 IFA>3THENA=0
60 M=P+M(A):IFM<SMORM>EMTHEN58
61 CP=PEEK(P):CM=PEEK(M):IFC>0ANDCP=210THENP=M:GOTO57
62 TM=M-SM:IF(CP=CMORCM<210)ANDC>0THEN58
63 ME=TM-L*INT(TM/L):IF(ME=0ANDM(A)=1)OR(ME=GANDM(A)=-1)THEN58
64 OD=INT(15/W(A)):CP=CP/W(A):POKEP,CP:CM=CM/OD:POKEM,CM
65 P=M:C=C+1:IFC<HTHEN57
66 RETURN
67 GOSUB90:PRINT"KARTE"
68 A=TL+LL+1:MS=A:DC=SM:DD=SM+G
69 FORB=DCTODD:FORC=0TO3:CB=PEEK(B):CB=CB-W(C)*INT(CB/W(C))
70 AA=1:IFC=1ORC=3THENAA=LL
71 IFC>1THENAA=-AA
72 BB=LL/AA:P2=A+AA:P1=P2+BB:P3=P2-BB
73 IFB=PTHENPOKEA,W4:POKEA+VP,CO
74 IFCB=0THENPOKEP1,W3:POKEP2,W3:POKEP3,W3
75 IFCB=0THENPOKEP1+VP,CO:POKEP2+VP,CO:POKEP3+VP,CO
76 NEXT:A=A+2:NEXT:DC=DC+L:DD=DD+L:A=MS+(2*LL):MS=A
77 IFDD<=EMTHEN69
78 GOSUB91:RETURN
79 REM
80 TL=7680:BL=8164:TR=TL+21:BR=BL+21:LL=22:DW=3:CD=3
81 W1=99:W2=100:W3=160:W4=88
82 W(0)=5:W(1)=7:W(2)=3:W(3)=2:L=INT(RND(1)*5+6)
83 M(0)=1:M(1)=L:M(2)=-1:M(3)=-L:M(4)=TL:M(5)=BL:M(6)=TR:M(7)=BR
84 W=INT(RND(1)*5+6):H=L*W-1:G=L-1:C=0:DC=0:T=0:RM=0
85 SM=7480:EM=SM+H:FORA=SMTOEM:POKEA,210:NEXT:P=SM+INT(RND(1)*H)
86 EC=SM+INT(RND(1)*L):MC=INT(RND(1)*H)+SM
87 OX=INT(H/3)+1:IFP<OXTHENOX=-OX
88 OX=OX+H:OL=1:TT=INT(RND(1)*H/8)+1:GOSUB56:GOSUB90
89 M=0:NP=P:CE=PEEK(EC):CE=CE/W(3):POKEEC,CE:GOTO48
90 PRINT"J":RETURN
91 FORX=0TOTL:NEXT:RETURN
92 GETA$:IFA$=""THEN92
93 RETURN

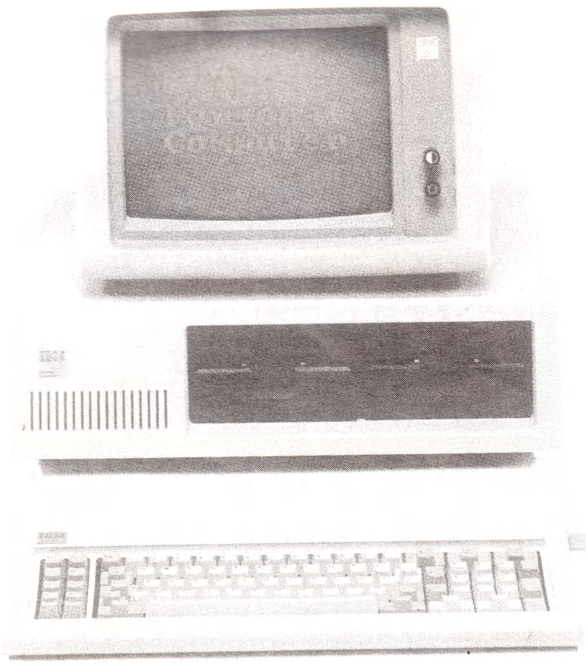
READY.
```

IBM PERSONAL COMPUTER

6 Monate Garantie

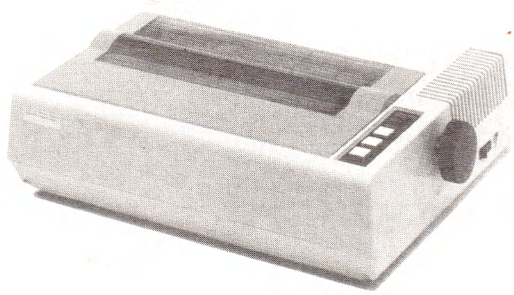
Computerversand
Albert-Schweitze
3436 Hessisch L

Alle genannten Preise incl. Mehrwe



Systemeinheit 64 KB + Disklaufwerk 320 KB	6590,--
Tastatur	870,--
Bildschirm	1110,--
Adapter für Drucker und Bildschirm	1090,--
Grafik Drucker	1980,--
Druckerkabel	185,--
2. Laufwerk 320 KB	1495,--
DOS	137,50
Speichererweiterung	950,--
Komplettes System (wie vor)	13300,--

Systemeinheit 64 KB + Disklaufwerk 160 KB	5950,--
Tastatur	870,--
Bildschirm	1110,--
Adapter für Drucker und Bildschirm	1090,--
DOS	137,50
Komplettes System (wie vor)	8690,--



H. Schädel
-Straße 1
ichtenau

rtsteuer + Versandkosten

olivetti – M 20 –

Der eingeführte bewährte Personal Computer mit den perfekten Problemlösungen



olivetti-Computer M 20
mit 160 KB
+ 2 Disklaufwerken à 320 KB 9950,--

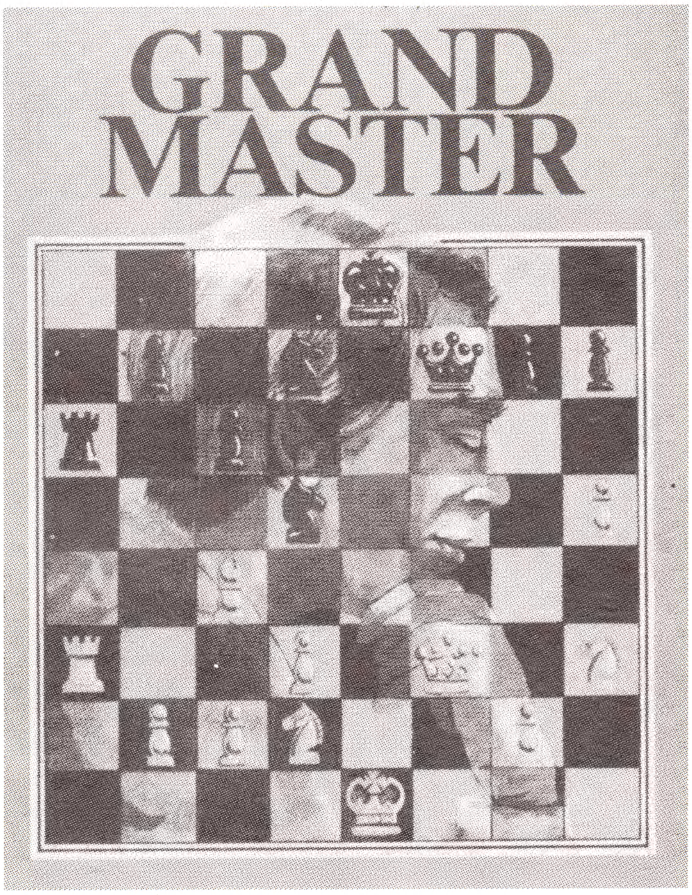


TI 99/4A

6 Monate Garantie

TI 99/4A 16 K RAM	699,--
Recorder	135,--
Kabel für Recorderanschluß	47,75
Sprachsynthesizer	349,--
Joysticks	95,--

Schachprogramm Grandmaster – ein Meister seines Fachs



Seit es Computer gibt, übte das „Königliche Spiel“ eine starke Faszination und Herausforderung auf viele Leute aus, die sich mit dem Programmieren beschäftigen. War es anfangs selbst auf großen Anlagen kaum möglich, den Computer zu einem guten und für den geübten Spieler ernsthaften Gegner zu entwickeln, so ist dies heute dank der rasanten Entwicklung der Mikroprozessortechnik selbst bei den Kleinen möglich geworden.

Aus der Fülle der angebotenen Schachprogramme, die inzwischen auf dem Markt sind, fiel uns das 1982 von der Firma „Kingsoft“ entwickelte Programm GRANDMASTER auf. Grandmaster gibt es als Cassette für den VC-20 und VC-64. Für die VC-20 Version ist eine Speichererweiterung um mindestens 8 K notwendig.

Kingsoft selbst bezeichnet sein Programm als das „stärkste Schachprogramm für Heimcomputer“. Diese Behauptung möchten wir an dieser Stelle weder bestätigen noch in Frage stellen. Nachdem wir Grandmaster jedoch ausführlich getestet haben, können wir sagen, daß dieses Programm in punkto Handling, Grafik und Spielstärke sehr ansprechend und last not least auch vom Preis (79,-- DM unverbindl.) recht interessant ist.
Die gutverständliche deutsche Bedienungsanleitung rundet das Programm ab.
Nach dem Laden des Programms waren wir von der Grafik und der übersichtlichen Anordnung der Bildschirmanzeigen äußerst angetan. Für die Ästhetiker unter den Schachspielern wurden einige reizvolle Möglichkeiten, den Bildschirm nach persönlichem Geschmack einzurichten, eingebaut. So kann die Grafik mittels der Cursortasten in horizontaler und vertikaler Richtung exakt auf die Bildschirmmitte zentriert werden. Über die Funktionstasten können das Spielbrett sowie Vorder- und Hintergrund nach Lust und Laune jede mögliche Farbkombination erhalten. Zugegeben, eine Spielerei, aber dennoch sehr schön!

Doch nun zum eigentlichen Spiel:
Es stehen 10 Spielstufen mit Rechenzeiten zwischen 5 Sekunden und 2 Stunden zur Verfügung. Für Freunde des Briefschachs ist Stufe Ø vorgesehen, bei welcher der Computer Stunden oder unter Umständen sogar Tage für seine Zugentscheidung benötigt.
Eine Änderung der Spielstärke ist zu jeder Zeit über die Taste “S“ in Verbindung mit der entsprechenden Zahl möglich.
Während Grandmaster überlegt, zeigt er den momentan favorisierten Zug und wieviel Halbzüge er im Voraus analysiert hat. Betätigen der “Stop-Taste“ unterbricht seine Überlegungen und veranlaßt ihn, diesen Zug auszuführen. Diese Funktion ist sehr praktisch, wenn auf einer

höheren Spielstufe gespielt wird und der Spieler nicht die endgültige Entscheidung des Computers abwarten möchte.
Die folgenden Möglichkeiten ergänzen die praktische Handhabung. Mit der Funktionstaste F2 kann auf automatisches Spiel umgeschaltet werden. Grandmaster spielt nun so lange gegen sich selbst, bis er ein Matt oder Patt findet. Für einen Anfänger zum Erlernen von Strategien ideal geeignet! Einfaches Betätigen der STOP-Taste führt wieder zum normalen Spiel zurück.
Um eine Figur zu setzen, wird lediglich die Start- und Zieladresse eingegeben (z. B. E2 E4) und RETURN gedrückt. Mittels der DEL-Taste können falsch eingegebene Zeichen gelöscht werden, wenn RETURN noch nicht gedrückt war. Auch nach dem RETURN-Befehl ist eine Zugrücknahme noch möglich und zwar über die ←-Taste. Für den ernsthaften Spieler ist dies natürlich Mogelei und daher nicht akzeptabel; für Anfänger jedoch unter Umständen lehrreich.
Grandmaster verfügt selbstverständlich über eine Eröffnungsbibliothek, die alle gängigen Varianten der Spieleröffnung enthält. CTRL Ø dreht das Spielbrett um; der Spieler kann mit Schwarz gegen den Computer spielen.
Alle Sonderzüge, Rochade, enPassant-Schlagen sowie die Bauernumwandlung sind möglich (Bauern werden grundsätzlich in eine Dame umgewandelt).

Unser Urteil: Ein sehr gutes, empfehlenswertes Programm, den meisten reinen Schachcomputern zumindest ebenbürtig, jedoch wesentlich preiswerter, als alles bisher angebotene.

CAR-CRASH

Ein Spielprogramm für den VC-20 mit mindestens 8K Speichererweiterung.
Spielen können zwei Spieler gegeneinander oder ein Spieler allein gegen den Computer.
Ziel des Spieles ist es, das gegen den Uhrzeigersinn fahrende Auto mit den Tasten "," und "." (innen, außen) möglichst lange unfallfrei zu fahren. Wenn fast alle Punkte verbraucht sind, erhält man ein "Bonusauto".
Beantwortet man die anfangs gestellte Frage nach der Spielerzahl mit "2", so kann ein zweiter Spieler das im Urzeigersinn fahrende Crashauto gleichzeitig mit "C=" (das ist die Commodore-Taste) und "Shift" (innen, außen) steuern.
Vor dem Laden oder Eintippen muß POKE44,32:POKE8182,Ø: NEW eingegeben werden.

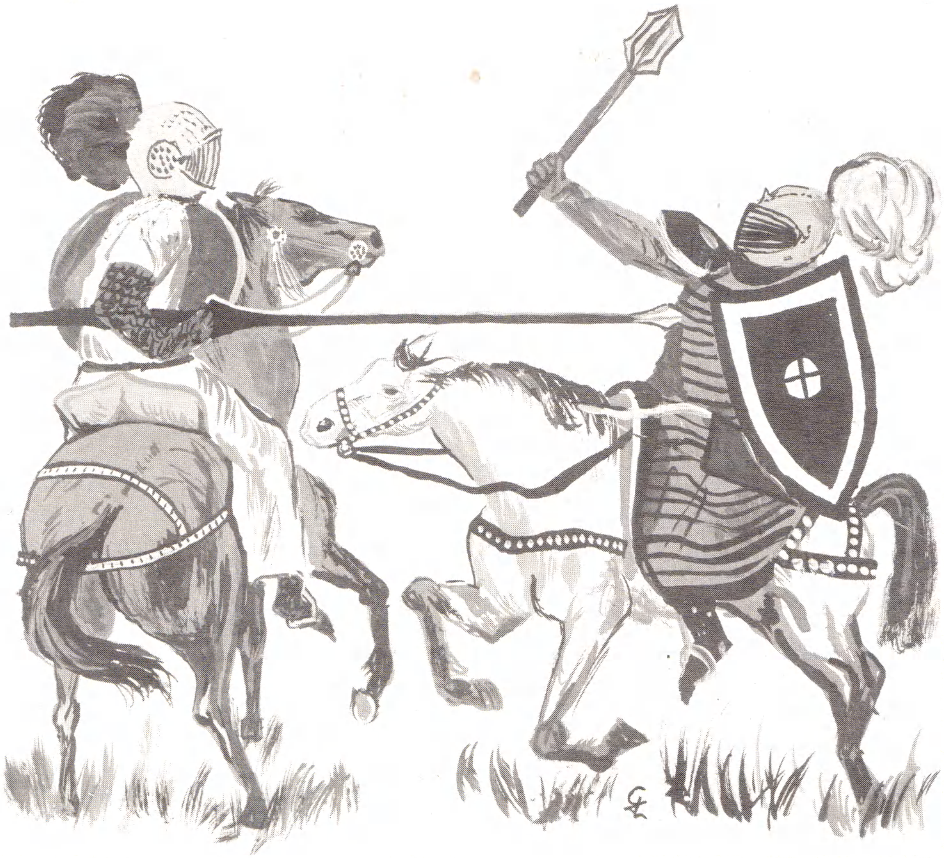
Georg Foltin

```
1 REM
2 REM CAR CRASH
3 REM
4 REM BY GEORG FOLTIN
5 REM
6 REM 1983
7 REM
8 REM LAENGE: 4.2K
9 REM
10 PRINT"■■■■"CHR$(8);:POKE36879,25
20 FORI=1TO21
25 POKE646,RND(1)*6+2:PRINT" ";
30 NEXT
40 PRINT"■■"
41 PRINTTAB(5)"■■           "
42 PRINTTAB(5)"■   VC=20    "
43 PRINTTAB(5)"■            "
44 PRINTTAB(5)"■ CAR CRASH  "
45 PRINTTAB(5)"■            "
50 PRINT"■■■";:FORI=1TO21
60 POKE646,RND(1)*6+2:PRINT" ";:NEXT
70 GOSUB7000
80 PRINT"■■":PRINT"■   (1) := 1 SPIELER"
85 PRINT:PRINT"    (2) := 2 SPIELER
86 GETA$:IFA$<>"1"ANDA$<>"2"THEN86
87 Y=VAL(A$+"")
88 PRINT"      ■■■■BITTE WARTEN"
100 GOSUB5000
110 REM BEWEGUNG MENSCH
120 POKEPM,32
130 AC=PEEK(PM+V(IM))
140 IFAC=81THENGOSUB3000:ONSGOTO110,200
150 IFAC=32THEN200
160 IFPC=PM+V(IM)THEN4000
170 IM=(IM+1)AND3
200 POKET,0:
205 PM=PM+V(IM)
210 POKEPM,C(IM)
215 IFPC=PM+V(IM)THEN4000
220 POKEPC,OC
230 AC=PEEK(PC+V(IC))
240 IFAC=81ORAC=32THEN300
250 IFPM=PC+V(IC)THEN4000
260 IC=(IC-1)AND3
300 PC=PC+V(IC)
```

Starship VC-20

```
310 OC=PEEK(PC):POKEPC,C(IC)
315 IFPM=PC+V(IC)THEN4000
320 V=PEEK(197):V=(V=37)-(V=29)
330 IFV=0THEN450
340 V=V*V((IM+1)AND3)
350 IFPEEK(PM+V)<>32THEN450
355 POKET+1,190:POKET,245
360 POKEPM,32:POKEPM+V,C(IM)
370 IFPM+2*V=PCTHEN4000
380 POKEPM+V,32:PM=PM+2*V
390 POKEPM,C(IM)
440 POKET+1,0:POKET,0
450 ONYGOTO451,460
451 IFRND(1)>.6THEN550
455 V=SGN(RND(1)-.5)
456 GOTO470
460 V=PEEK(653):V=(V=2)-(V=1)
465 IFV=0THEN550
470 V=V*V((IC+1)AND3)
480 IFPEEK(V+PC)<>32THEN550
490 POKEPC,32:POKEPC+V,C(IC)
500 IFPC+2*V=PMTHEN4000
510 POKEPC+V,32:PC=PC+2*V
520 POKEPC,C(IM)
550 GOTO110
3000 S=2
3010 SC=SC+1:POKET,230:POKET-1,0
3020 PRINTP$"■"MID$(STR$(SC),2)
3030 ZZ=ZZ+1
3040 IFZZ<140THENRETURN
3042 FORI=1TO9:POKET,220:POKET+3,76
3044 FORQ=0TO99:NEXT
3046 POKET,0:POKET+3,25:FORQ=0TO99:NEXT:NEXT
3050 ZZ=0:S=1:XX=XX+1
3055 POKET,0
3060 GOSUB5050
3070 PRINTP$;"◣▣+";XX-1
3080 RETURN
4000 POKEPC,32:POKEPM,32:POKET+1,0
4010 FORI=10TO100STEP10
4015 POKET+1,IOR128
4020 FORJ=0TO3
4030 POKEPC,C(J):POKEPC+33792,RND(1)*6+2
4035 FORTT=0TOI:NEXT
4040 NEXT:NEXT
4045 POKET+1,0:POKEPC,32:POKEPC+33792,6
4050 XX=XX-1:IFXX>0THENPM=4503:IM=0:PC=4295:IC=1:OC=32:PRINTP$;"▣◣+";XX-1:GOTO11
0
4060 POKE198,0
4065 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣"
4070 P=4502:I=0:F=0
4075 POKEP,C(I):C=PEEK(P+V(I))
4080 IFC=64ORC=93THENI=(I+1)AND3:F=(F+1)AND7:IFF<>6THENPOKE36879,17*F+8
4085 FORTT=0TO40:NEXT
4090 POKEP,32:P=P+V(I)
4095 GETA$:IFA$=""THEN4075
4100 GOSUB5045:POKE36879,25
4110 GOTO110
5000 FORI=0TO511:POKEI+6144,PEEK(I+32768):NEXT
5020 READC:IFC=-1THEN5045
5030 A=6144+8*C
5040 FORI=0TO7:READC:POKEI+A,C:NEXT:GOTO5020
5045 XX=3:SC=0:ZZ=0
5050 POKE36869,206:POKE36879,25:POKE36878,15:T=36876
5055 PM=4503:IM=0:PC=4295:IC=1:OC=32:P$="▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣"
5060 PRINT"▣▣";
5065 C(0)=107:C(1)=113:C(2)=115:C(3)=114:V(0)=1:V(1)=-22:V(2)=-1:V(3)=22
5070 POKE650,128
5080 PRINT"┌──────────────────┐";
5090 PRINT"|●●●●●●●●●   ●●●●●●●●●|";
5100 PRINT"|● ┌──────┐ ┌──────┐●|";
5110 PRINT"|●|●●●●●●●   ●●●●●●●|●|";
5120 PRINT"|●|● ┌────┐ ┌────┐●|●|";
5130 PRINT"|●|●|●●●●●   ●●●●●|●|●|";
5140 PRINT"|●|●|● ┌──┐ ┌──┐●|●|●|";
5150 PRINT"|●|●|●|●●●   ●●●|●|●|●|";
5160 PRINT"|●"●"●"● ┌──────┐●"●"●"●|";
5170 PRINT"|        |      |       |";
5180 PRINT"|        |+ 2   |       |";
5190 PRINT"|●"●"●"● └──────┘●"●"●"●|";
5200 PRINT"|●|●|●|●●●   ●●●|●|●|●|";
5210 PRINT"|●|●|● └──┘ └──┘●|●|●|";
5220 PRINT"|●|●|●●●●●   ●●●●●|●|●|";
5230 PRINT"|●|● └────┘ └────┘●|●|";
5240 PRINT"|●|●●●●●●●   ●●●●●●●|●|";
5250 PRINT"|● └──────┘ └──────┘●|";
5260 PRINT"|●●●●●●●●●   ●●●●●●●●●|";
5270 PRINT"└──────────────────┘";
5280 PRINT"▣▣      ▣CAR ▣CRASH▣"
5300 POKEPM,C(IM):POKEPC,C(IC)
5310 RETURN
7000 POKE36878,15:FORI=0TO1
7010 RESTORE
7020 FORJ=1TO22
7030 READH,L
7040 POKE36875+I,H
7050 FORK=1TO300/L
7060 NEXTK
7070 POKE36875+I,0
7080 FORK=1TO10
7090 NEXTK
7100 NEXTJ
7110 NEXTI
7999 RETURN
9046 POKET,0:POKET+3,25:FORQ=0TO99:NEXT:NEXT
10000 DATA201,2,191,2,201,4,201,4,191,2,183,4,183,4,175,4,183,4,191,2,175,2,201,
2
10010 DATA191,2,201,4,201,4,191,2,183,4,183,4,183,4,191,4,175,2,175,2
20000 DATA64,0,0,255,0,0,255,0,0
20010 DATA93,36,36,36,36,36,36,36,36
20020 DATA75,36,36,196,4,8,240,0,0
20050 DATA108,0,0,240,8,8,240,0,0
20060 DATA74,36,36,35,32,16,15,0,0
20070 DATA73,0,0,240,8,4,196,36,36
20080 DATA85,0,0,15,16,32,35,36,36
20100 DATA123,0,0,15,16,16,15,0,0
20110 DATA126,36,36,36,36,24,0,0,0
20120 DATA124,0,0,0,24,36,36,36,36
20130 DATA113,24,90,126,90,24,219,255,219
20140 DATA114,219,255,219,24,90,126,90,24
20150 DATA107,224,238,68,255,255,68,238,224
20160 DATA115,7,119,34,255,255,34,119,7
20170 DATA81,0,0,0,24,24,0,0,0
20180 DATA48,28,51,51,0,51,51,28,0
20190 DATA49,0,3,3,0,3,3,0,0
20200 DATA50,28,3,3,28,48,48,28,0
20210 DATA51,28,3,3,28,3,3,28,0
20220 DATA52,0,51,51,28,3,3,0,0
20230 DATA53,28,48,48,28,3,3,28,0
20240 DATA54,28,48,48,28,51,51,28,0
20250 DATA55,28,3,3,0,3,3,0,0
20260 DATA56,28,51,51,28,51,51,28,0
20270 DATA57,28,51,51,28,3,3,28,0
20280 DATA84,255,255,24,24,24,24,24,24
20290 DATA69,255,255,192,255,255,192,255,255
20300 DATA72,195,195,195,255,255,195,195,195
20310 DATA68,252,252,195,195,195,195,252,252
20320 DATA78,195,195,243,243,207,207,195,195
20330 DATA96,170,85,170,85,170,85,170,85,-1
30000 REM (C) BY
30001 REM GEORG FOLTIN
30002 REM    &
30003 REM CORD KIELHORN
30004 REM 1983

READY.
```



Ritter

Ihre Aufgabe ist es eine Burg einzunehmen. Sie können sich mit den Tasten:

1 ← links

E → rechts

P Schild

L Schuß bewegen.

Sinn des Spieles ist es, die Kette der Zugbrücke durchzuschießen, indem Sie ganz nach vorne zum Graben gehen und dann auf „Teufel komm raus“ losfeuern.
Es ist natürlich klar, daß sich die Burgbewohner so etwas nicht gefallen lassen. Sie werden sich mit Bogenschützen und Wunderwolken verteidigen.
Wenn ein Pfeil auf Sie zufliegt, müssen Sie sich ducken „P“. Bei der Wunderwolke schützen Sie sich entweder mit dem Schutzschild „P“ vor dem Pechregen, aber besser ist es ihm auszuweichen.
Sie sollen die Kettenbefestigung 16 mal treffen. Wenn Sie sich ducken oder das Schutzschild benutzen, wird Ihnen jeweils 1 – 20 Treffer abgezogen.

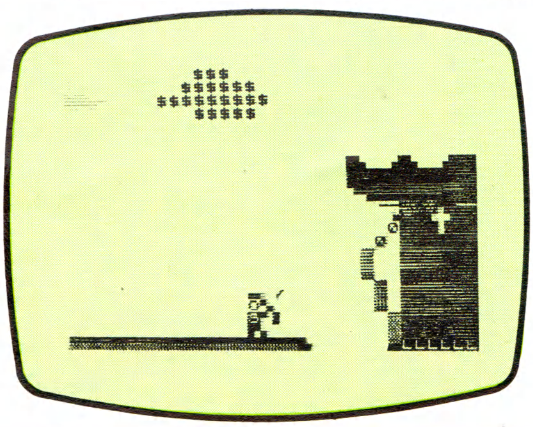
Sie fangen mit 0 Treffern an.
Sie sollen 16 Treffer erreichen und können maximal – 6 Treffer durch ducken oder Schutzschildbetätigung bekommen. Jedesmal wenn Sie sich ducken wird nämlich die Kette erneuert bzw. repariert.

Zum Programm:
Als erstes geben Sie ein:
1 Rem . . . (19 . Punkte . ein). N/L
Danach schreiben Sie
Poke 16510,0 N/L
und dann geben Sie das Listing wie gewohnt der Reihe nach ein.

Bemerkungen:
Bevor Sie Ihren ersten Probelauf machen, saven Sie lieber das Programm, denn wenn in den ersten 7 Zeilen ein Fehler ist, kann dann das Programm gelöscht werden, wenn das MC-Programm in Zeile 560 und 4010 abgerufen wird.

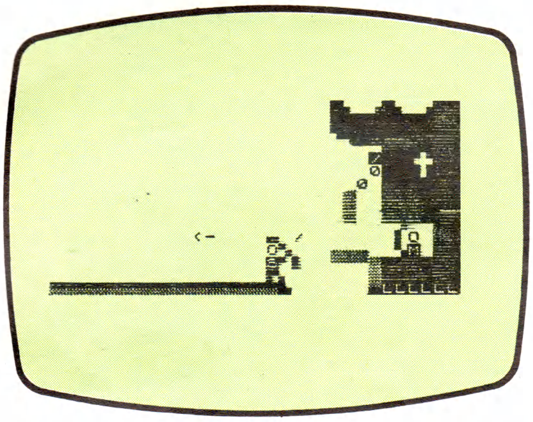
In Zeile 510 wird man in den Gefahrenabschnitt gebracht. Durch verändern des Wertes hinter dem „RND“ beeinflußt man die Gefahren.
In Zeile 570 wird bestimmt, ob man dem Bogenschützen oder der Wolke ausgesetzt wird.
In der Zeile 420 werden die Inputzeilen zu den Ausdruckzeilen dazugegeben (Bildschirm) (24-Zeilen-Bildschirm)

Stefan Benk

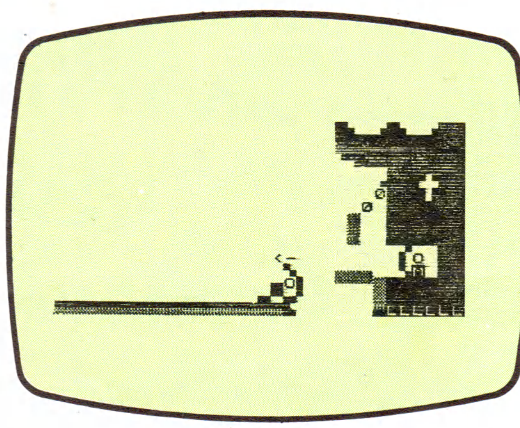


```
   0 REM E£RND:/ 477$4 NEXT TAN
   1 PRINT AT 0,0;"    EROBERN SI
E EINE BURG...                GE
STEUERT WIRD MIT:
                     1.........L
INKS                 E.........RE
CHTS                 L.........SC
HUSS                 P.........SC
HUTZ",,TAB 3;"VIEL GLUECK"
   2 RAND
   3 LET A$="A2C004E081600232E7E
E0877019F32D0023FSC"
   4 FOR A=16514 TO 16532
   5 POKE A,CODE A$+16*CODE A$(2
)-476
   6 LET A$=A$(3 TO )
   7 NEXT A
   8 FOR A=1 TO 50
   9 NEXT A
  10 LET A$="
  20 LET A$=A$+"
  30 LET A$=A$+"
  40 LET A$=A$+"
  50 LET A$=A$+"
  60 LET A$=A$+"
  70 LET A$=A$+"
  80 LET A$=A$+"
  90 LET A$=A$+"
 100 LET A$=A$+"
 110 LET A$=A$+"
 120 LET A$=A$+"
 130 LET A$=A$+"
 140 LET A$=A$+"
 150 LET A$=A$+"
 160 LET A$=A$+"
 170 LET A$=A$+"
 180 REM "        /"
 190 REM " O       "
 200 REM "         "
 210 REM "         "
 220 REM
 230 REM "         "
 240 REM "  O      "
 250 REM "         "
 260 REM "         "
 270 REM
 280 REM "         "
 290 REM "O        "
 300 REM "       - "
 310 REM "         "
 320 REM
 330 REM "     O   "
 340 REM "         "
 350 REM
 360 REM "       O "
 370 REM "         "
 380 REM
 390 REM "       O "
 400 REM "         "
 410 REM
 420 POKE 16418,0
 425 CLS
 430 PRINT AT 7,0;A$;AT 17,3;"
 /";TAB 3;"    O";TAB 3;"     ";
TAB 3;"   "
 440 LET A=4
 450 LET L=0
 460 LET S=0
 470 LET D=0
 480 LET B=1
 490 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
 500 PRINT AT 17,A;"  / ";TAB A
;" O  ";TAB A;"    ";TAB A;"
"
 510 IF RND<.15 THEN GOSUB 550
 520 LET D=INKEY$="L"
 530 IF D OR S THEN GOSUB 1080
 540 GOTO 490
 550 REM     ACHTUNG
 560 PRINT TAB USR 16514;AT 22,2
;"ACHTUNG...";TAB USR 16514
 570 IF RND>.5 THEN GOTO 730
 580 REM    BOGENSCHUETZE
 590 PRINT AT 16,24;"        ";TAB
24;"  <- O  ";TAB 22;"        ";T
AB 22;"
 600 FOR C=31 TO 0 STEP -2
 610 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
 620 IF B THEN PRINT AT 17,A;"
 /  ";TAB A;" O  ";TAB A;"    ";
TAB A;"   "
 630 IF NOT B THEN PRINT AT 17,A
;"      ";TAB A;"      ";TAB A;"  O
   ";TAB A-1;"
 640 LET B=INKEY$<>"P"
 650 IF NOT B THEN LET L=L-(L>-6
)
 655 LET D=INKEY$="L"
 660 IF D OR S THEN GOSUB 1080
 670 IF C<25 THEN PRINT AT 17,C;
"<-  "
 680 IF ABS (A-C+1)<2 AND B THEN
GOTO 2000
 690 NEXT C
 700 PRINT AT 17,0;"       "
 710 PRINT AT 22,2;"           ";
AT 16,24;"        ";TAB 24;"
";TAB 22;"         ";TAB 22;"
 720 RETURN
 730 REM     PECH
 740 FOR C=22 TO A-2 AND A>3 STE
P -1
 750 PRINT AT 0,C+3;"$$$  ";TAB C
+2;"$$$$$$  ";TAB C;"$$$$$$$$  ";
TAB C+3;"$$$$$  "
 760 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
 770 LET B=INKEY$<>"P"
 775 IF NOT B THEN LET L=L-(L>-6
)
 780 IF B THEN PRINT AT 17,A;"
 /  ";TAB A;" O  ";TAB A;"    ";
TAB A;"   "
 790 IF NOT B THEN PRINT AT 17,A
-1;"      ";TAB A-1;"  O    ";TAB
A;"    ";TAB A;"   "
 795 LET D=INKEY$="L"
 800 IF D OR S THEN GOSUB 1080
 810 NEXT C
 820 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;"   ";TAB 17;"    ",TAB
18;"  ";TAB 17;" "
 830 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;"   ";TAB 17;"    ",TAB
18;"  ";TAB 17;" "
 840 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;"   ";TAB 17;"    ",TAB
18;"  ";TAB 17;" "
 850 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;"   ";TAB 17;"    ",TAB
18;"  ";TAB 17;" "
 860 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;"   ";TAB 17;"    ",TAB
18;"  ";TAB 17;" "
 870 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;"   ";TAB 17;"    ",TAB
18;"  ";TAB 17;" "
 880 PRINT AT 0,C+3;"      ";TAB
C+2;"       ";TAB C+1;"
```

```
";TAB C+3;"■       "
 881 PRINT AT 0,C+3;"    ";TAB C+2
;"  ■■   ";TAB C+1;"      ■   ";T
AB C+3;"■"
 882 PRINT AT 1,C+2;"        ";TAB
C+1;"    ■■■   ";TAB C+3;"■    "
 883 PRINT AT 1,C+2;"        ";TAB
C+1;"         ";TAB C+3;"■■■■"
 884 PRINT AT 3,C+3;"        ";TAB C
+3;" ■■ "
 885 PRINT AT 4,C+3;"        ";TAB C
+3;" ■ "
 890 FOR Z=3 TO 18 STEP 2
 900 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
 910 LET D=INKEY$="L"
 920 IF D OR S THEN GOSUB 1080
 930 LET B=INKEY$<>"P"
 940 IF B THEN PRINT AT 17,A;" ■
/ ";TAB A;" O■ ";TAB A;" ■■ ";
TAB A;" ■ "
 950 IF NOT B THEN PRINT AT 17,A
-1;" ■■■ ";TAB A-1;" ■O■ ";TAB
A;" ■ ";TAB A;" ■ "
 960 IF NOT B THEN LET L=L-(L>-6
)
```



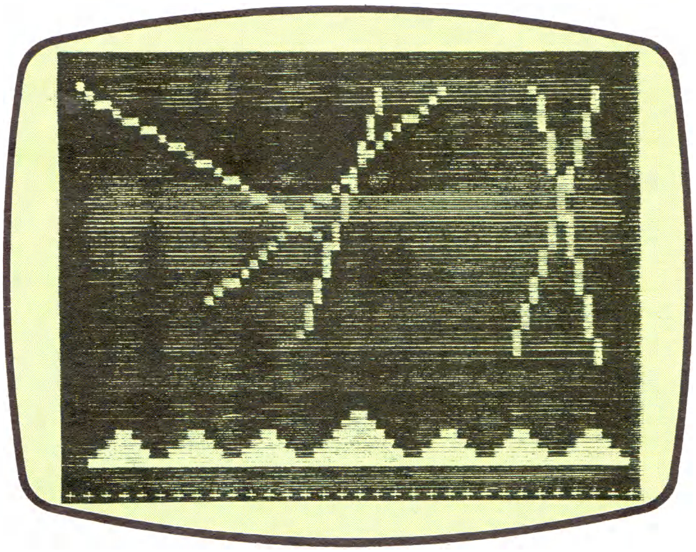
```
 970 PRINT AT Z,C+4;"■";AT Z-2,C
+4,
 980 NEXT Z
 990 PRINT AT Z-2,C+4,
 995 PRINT AT 22,0;"■■■■■■■■■■■■
■■"
1000 IF ABS (A-C-3)>1.6 OR NOT B
 THEN RETURN
1005 LET L=RND**RND**RND
1010 PRINT AT 17,A-1;"   ■   ";TA
B A-1;" ■■■■ ";TAB A-1;"■■■■ ";
TAB A-1;"  ■■  ";TAB A;"■ ■■";
1015 LET L=RND**RND**RND
1020 PRINT AT 17,A;"        ";TAB A-
1;"       ";TAB A-1;" ■■■ ";TAB
A-1;"■■■■ ";TAB A-1;"■■M■■";TA
B A;"■■■■";
1025 LET L=RND**RND**RND
1026 PRINT AT 18,A;"        ";TAB A-
1;"       ";TAB A-1;" ■■$■ ";TAB
A-1;"■■■■■";TAB A-1;"■■M■■";TA
B A;"■■■■";
1027 LET L=RND**RND**RND
1030 PRINT AT 0,0;"DER ""PECH-RE
BEN"" HAT SIE                ER
SCHLAGEN■■■■■■■■■■"
1040 GOTO 2050
1080 IF S THEN GOTO 1140
1090 LET H=18
1100 FOR S=A+3 TO A+9
1110 LET H=H-1
1120 PRINT AT H+1,S-1,AT H,S;"/"
;
1130 RETURN
1140 NEXT S
1150 PRINT AT H,S-1,
1160 LET L=L+(S=26)
1170 IF S=26 THEN PRINT AT 11,25
;CHR$ (L+156)
1180 LET S=0
1190 IF L=16 THEN GOTO 4000
1200 LET D=0
1210 RETURN
2000 PRINT AT 17,A;" ■ / ";TAB A
;" O■ ";TAB A;" ■■- ";TAB A;" ■
■"
2005 LET L=RND**RND**RND
2010 PRINT AT 17,A;" ■ / ";TAB A
;" O■ ";TAB A;" ■<■ ";TAB A;" ■■
"
2015 LET L=RND**RND**RND
2020 PRINT AT 17,A;" ■ / ";TAB A
;" O■ ";TAB A;" ■-■ ";TAB A;" ■■
"
2025 LET L=RND**RND**RND
2030 PRINT AT 17,A;"        ";TAB A
;"  ";TAB A;"/     ";TAB A-1;"O
■■ "
2035 LET L=RND**RND**RND
2040 PRINT AT 0,0;"VOM BOGENSCHU
ETZEN ERWISCHT"
2050 PRINT AT 10,0;"NOCHMAL?"
2060 RUN (2060 AND INKEY$<>"J")
4000 FOR Z=0 TO 30
4010 LET L=USR 16514+USR 16514
4020 NEXT Z
4030 PRINT AT 22,1;"GEWONNEN...."
;TAB 0;"■■■■■■■■■■■■■■■■■"
4040 FOR Z=14 TO 20
4050 PRINT AT Z,23;"     ";
4060 NEXT Z
4070 PRINT AT 20,19;"        "
4080 PRINT AT 21,18;"■■■■■■■■■■"
4090 FOR A=A TO 25
4100 PRINT AT 17,A;" ■ / ";TAB A
;" O■ ";TAB A;" ■■ ";TAB A;" ■■
"
4110 NEXT A
4115 PRINT AT 1,0;"HERZLICHEN GL
UECKWUNSCH           HERR RITTE
R"
4120 PRINT AT 7,0;A$( TO LEN A$-
64)
4130 PRINT AT 10,0;"NOCHMAL?"
4140 RUN (4150 AND INKEY$<>"J")
4150 PRINT AT 6,22-6;"■JUHU■)■O■
```



```
";TAB 22;"■■S■■";
4155 LET L=RND**RND
4160 PRINT AT 6,22-6;"          O ■
";TAB 22;"■■S■■";
4170 PRINT AT 6,22-6;"           O
";TAB 22;"■■S■■";
4180 PRINT AT 6,22-6;"           O
";TAB 22;"■■S■■";
4190 PRINT AT 6,22-6;"            O
";TAB 22;"■■S■■";
4200 PRINT AT 6,22-6;"          ■O
";TAB 22;"■■S■■";
4210 GOTO 4140
9940 STOP
9950 SAVE "RITTE■"
9960 RUN
```

Missile ZX-Command

In Mußestunden habe ich versucht, das bekannte Spielhöllenspiel "Missile Command" so gut es mir möglich war, auf dem ZX-81 zu gestalten.



Zur Eingabe:
Wenig angetan von der fehlerträchtigen Eingabeart eines Hex-Dumps in C$, habe ich durch Ausdruck der dezimalen Äquivalente und einer entsprechenden Eingaberoutine (Zeilen 8000-8004) den Versuch gemacht, sowohl die Eingabe schneller, als auch die Fehlerrate (z.B. Verwechselung von B und 8) geringer zu machen. Das hat obendrein den Vorteil, daß keine Initialisierung notwendig ist.
Nach einmaliger Eingabe der dezimalen Liste (von links nach rechts!) kann die Eingabe-Routine gelöscht werden, da die REM-Zeile ja mitgesaved wird.

Beim ersten Start des Programms mit RUN muß zuerst die dezimale Datentabelle eingetippt werden. Bevor man dann das Programm ein für alle mal absaven kann, müssen die Zeilen 8000 - 8004 gelöscht werden. Jeder weitere Start erfolgt dann mit RUN.

Zum Spiel:
Die Aufgabe des Spielers ist es, möglichst lange die 6 Städte vor den Laserstrahlen zu schützen. Das kann er dadurch erreichen, daß er den grauen Zielbalken unter einen der Strahlen bewegt und den Feuerknopf betätigt.
An der Stelle des Zielbalkens wird dann eine Explosion entstehen, die den Strahl zerstört. Bis zu 3 Explosionen können gleichzeitig auf dem Bildschirm sein.
Es steht nur eine begrenzte Anzahl an Schüssen zur Verfügung. Sie werden unten als Pluspunkte dargestellt. Zusätzlich geht Energie verloren, wenn ein Strahl die Basisstadt in der Mitte trifft. Nach 50 Strahlen erfolgt eine Zwischenwertung. Das Spiel wird dann automatisch schneller fortgesetzt.
In der Version, die durch das Listing angegeben wird, haben die Tasten folgende Bedeutungen (Ziffern links).

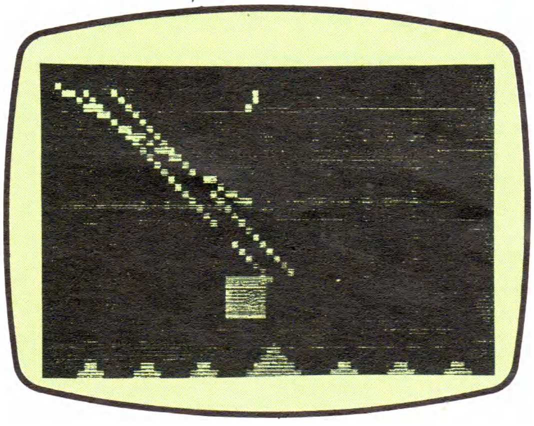
6	Zielbalken runter	D
7	Zielbalken rauf	O
8	Zielbalken links	W
9	Zielbalken rechts	K
0	Feuer	0

Die Buchstaben (rechts) habe ich so vereinbart, weil ich parallel zu diesen Tasten einen ATARI-Joystick angeschlossen habe. Für ZX-Besitzer, die das nicht haben, wird durch folgende Befehle eine wesentlich besser zu greifende Belegung erreicht:
POKE 16561,117
POKE 16556,85
POKE 16566,109
POKE 16571,77

Martin Zwerschke

```
   1 REM 1175 ZEICHEN (37     ZEI
LEN
   2 REM MISSILE COMMAND BY
         MARTIN ZWERSCHKE   3/83
  10 REM
  20 GOTO 8000
1000 PRINT AT 9,9;"
";TAB 9;" ";TAB 21;" ";TAB 9;"
GAME OVER ";TAB 9;" ";TAB 21;"
";TAB 9;"           "
1010 IF INKEY$="" THEN GOTO 1010
1020 RUN
2130 LET S=26033
2140 LET A$="111211231312233212З
1221144"
2200 FOR I=1 TO LEN A$
2220 LET S=S+1
2240 POKE S,VAL A$(I)
2260 NEXT I
2270 POKE S,0
2280 LET A=PEEK 26000
2290 LET B=PEEK 26001
2300 POKE 26022,A
2320 POKE 26023,B
2340 POKE 26026,A
2360 POKE 26027,B
2380 POKE 26030,A
2400 POKE 26031,B
3000 LET L=USR 16514
3100 LET L=USR 17533
3205 PRINT AT 0,0
3210 FOR I=0 TO 18
3220 PRINT "
"
3230 NEXT I
3240 IF PEEK 27030>3 THEN POKE 2
7030,PEEK 27030-3
3300 LET PUNKTE=PEEK 27024+256*P
EEK 27025
3310 PRINT AT 22,15;PUNKTE
3320 LET DF=PEEK 16396+256*PEEK
16397
3330 LET STAEDTE=0
3340 LET B$=""
3350 FOR I=DF+693 TO DF+792
3360 IF PEEK I<>118 THEN LET B$=
B$+CHR$ PEEK I
3370 NEXT I
3380 LET I=4
3382 GOSUB 4E3
3384 LET I=8
3386 GOSUB 4E3
3388 LET I=12
3390 GOSUB 4E3
3392 LET I=22
3394 GOSUB 4E3
3396 LET I=26
3398 GOSUB 4E3
3400 LET I=30
```

```
3402 GOSUB 4E3
3420 FOR I=65 TO 96
3430 IF B$(I)="█" THEN GOTO 5000
3435 GOSUB 3600
3440 NEXT I
3450 GOTO 5000
3600 PRINT AT 23,I-65;"█"
3610 LET PUNKTE=PUNKTE+1
3620 PRINT AT 22,15;PUNKTE
3630 RETURN
4000 IF B$(I)<>" " THEN RETURN
4010 PRINT AT 21,I-3;"███████";AT
22,I-3;"▒▒▒▒▒▒▒"
4020 LET PUNKTE=PUNKTE+10
4030 LET STAEDTE=STAEDTE+1
4032 FOR I=1 TO 14
4034 NEXT I
4036 PRINT AT 22,15;PUNKTE
4040 RETURN
5000 LET ABF=1
5001 IF STAEDTE=0 THEN GOTO 1000
5002 PRINT AT 21,0;B$
5004 POKE 27025,INT (PUNKTE/256)
5006 POKE 27024,PUNKTE-256*PEEK
27025
5008 PRINT AT 22,15;PUNKTE
5010 GOTO 9000
8000 FOR I=16514 TO 17689
8001 INPUT A
8002 POKE I,A
8003 PRINT I,PEEK I
8004 NEXT I
8008 POKE 16418,2
8009 CLS
8010 POKE 27024,0
8011 POKE 27025,0
8021 PRINT "STRAHLEN? (4/5)"
8022 INPUT STR
8023 POKE 27034,STR=5
8024 PRINT ,,"GESCHWINDIGKEIT ?
(1-10)"
8025 INPUT G
8026 IF G<1 OR G>10 THEN GOTO 80
24
8027 POKE 27030,102-10*G
8030 CLS
8999 GOTO 9998
9000 FOR I=27600 TO 27625
9005 POKE I,0
9020 NEXT I
9030 POKE 26012,1
9035 POKE 16418,0
9039 IF ABF THEN GOTO 2000
9040 PRINT AT 21,0;"▀▀▀   ▀▀▀   ▀▀▀
▀▀   ▀▀   ▀▀   ▀   ▀"
9042 PRINT AT 22,0;"▀▀
▀   ▀   ▀"
9044 PRINT "████████████████████████
███████████████"
9046 PRINT AT 20,0;"████████████
██████"
9997 GOTO 2000
9998 LET ABF=0
9999 GOTO 9000
```

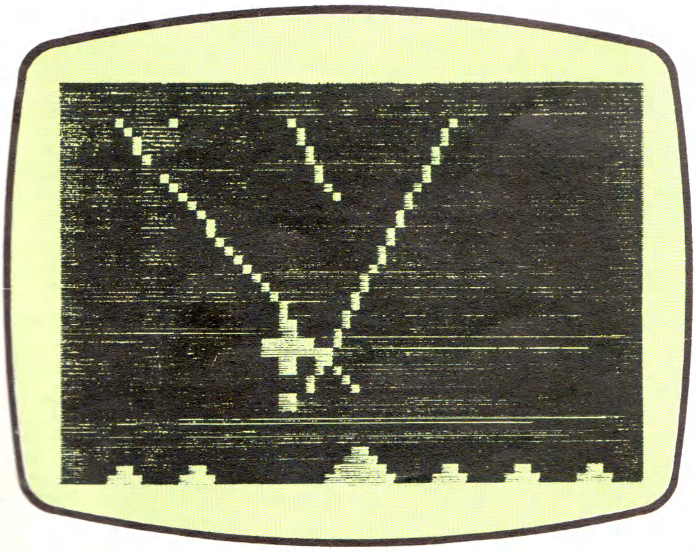


DATEN :

```
33   208  101  34   168  101  34   172
101  34   176  101  52   0    50   152
101  42   12   54   1    56   0    9
34   148  101  1    24   1    9    34
144  101  9    34   146  101  205  187
2    203  92   204  192  64   203  84
204  206  64   203  100  204  220  64
203  108  204  246  64   201  205  14
65   43   126  254  118  194  53   65
35   195  53   65   205  14   65   35
126  254  118  194  53   65   43   195
53   65   205  14   65   1    33   0
175  237  66   237  91   148  101  175
237  82   250  241  64   24   1    9
25   195  53   65   205  14   65   1
33   0    9    237  91   146  101  175
237  82   250  10   65   175  237  66
25   195  53   65   34   154  101  42
144  101  126  254  138  40   3    54
128  201  58   152  101  119  201  53
65   1    33   0    9    237  91   146
101  175  237  82   250  52   65   175
237  66   25   34   144  101  126  50
152  101  54   138  42   154  101  201
42   37   64   203  76   204  11   65
42   12   54   1    149  2    62   0
237  177  224  43   54   128  35   24
247  42   166  101  237  75   168  101
10   167  40   1    3    237  67   168
101  205  149  65   42   170  101  237
75   172  101  10   167  40   1    3
237  67   172  101  205  149  65   42
174  101  237  75   176  101  10   167
40   1    3    237  67   176  101  205
149  65   201  200  254  1    40   98
254  2    40   62   254  4    40   93
22   0    114  1    34   0    175  237
66   205  3    66   40   1    114  35
114  35   205  3    66   40   1    114
1    31   0    9    205  3    66   40
1    114  35   35   205  3    66   40
1    114  9    205  3    66   40   1
114  35   114  35   205  3    66   200
114  201  54   0    43   205  3    66
40   2    54   0    1    32   0    175
237  56   54   0    9    35   35   205
3    66   40   2    54   0    9    54
0    201  54   0    201  22   128  24
161  126  254  118  192  62   0    167
201  58   146  105  167  250  52   66
61   50   146  105  42   168  101  1
166  101  126  167  40   23   42   172
101  1    170  101  126  167  40   13
42   176  101  1    174  101  126  167
40   3    219  255  201  42   144  101
125  2    3    124  2    3    33   101
178  124  2    3    125  2    201  205
80   66   50   108  107  201  42   120
105  237  91   122  105  123  149  79
122  148  71   40   4    23   159  203
199  50   125  105  50   127  105  121
167  40   4    23   159  203  199  50
124  105  50   126  105  121  203  127
40   2    237  68   95   120  203  127
40   2    237  68   87   147  56   9
175  50   126  105  123  90   87   24
4    175  50   127  105  75   203  57
58   119  105  71   4    121  130  79
147  56   13   79   58   124  105  133
111  58   125  105  132  103  24   10
58   126  105  133  111  58   127  105
132  103  16   225  229  68   77   62
1    203  40   56   2    62   4    203
41   48   1    7    245  4    42   16
64   17   223  255  25   16   253  9
125  95   7    254  16   48   6    15
48   2    238  143  71   241  229  33
156  101  203  70   32   3    176  24
2    47   160  225  254  8    56   2
238  143  119  193  58   123  105  184
32   3    62   2    201  123  167  40
3    62   0    201  62   1    201  33
208  107  17   119  105  1    5    0
237  176  24   2    24   244  34   140
```

Die ZX-Seiten

105	43	43	43	43	43	126	167
200	52	205	73	66	254	1	202
230	67	254	2	202	239	67	42
140	105	24	214	14	0	58	208
107	167	40	38	14	5	58	213
107	167	40	30	14	10	53	218
107	167	40	22	14	15	58	223
107	167	40	14	58	154	105	167
200	14	20	58	223	107	167	40
1	201	33	208	107	6	0	9
54	1	35	229	205	143	67	62
60	165	225	119	35	54	45	35
229	205	143	67	62	60	165	225
119	35	54	9	201	42	50	64
17	77	0	124	181	40	34	68
77	62	16	33	0	0	41	203
17	203	16	48	4	25	48	1
3	61	32	242	124	230	128	176
177	167	237	66	48	5	35	24
2	237	82	34	50	64	201	33
156	101	203	134	42	140	105	1
5	0	175	237	66	17	119	105
229	237	176	205	73	66	225	53
32	229	33	156	101	203	198	33
148	105	52	201	42	144	105	35
34	144	105	24	210	42	140	105
43	43	126	203	63	38	3	254
4	40	36	30	7	254	8	40
30	30	11	254	12	40	24	254
16	40	73	30	21	254	20	40
14	30	25	254	24	40	8	30
29	254	28	40	2	24	160	42
12	64	1	181	2	9	22	0
25	54	128	35	54	128	35	54
128	1	30	0	9	126	254	3
54	128	32	2	54	3	35	54
137	35	54	137	35	54	137	35
126	254	3	54	128	32	2	54
3	195	193	67	58	146	105	167
250	193	67	14	10	145	50	146
105	195	193	67	33	156	101	203
22	63	62	128	56	2	62	155
203	30	42	12	64	1	132	2
9	119	201	33	148	105	54	0
62	96	50	146	105	62	127	219
254	31	208	58	148	105	254	50
200	205	168	64	205	66	55	205
91	65	58	150	105	203	63	71
61	167	32	252	62	250	16	248
205	62	67	205	168	64	205	17
67	205	102	58	205	168	64	205
207	68	205	91	65	58	150	105
203	63	71	61	167	32	252	62
250	16	248	24	184	42	12	64
1	248	2	9	58	146	105	50
152	105	6	32	58	152	105	214
3	50	152	105	250	237	68	54
149	24	2	54	128	35	16	236
201	0	0	28	28	28	28	28
28	26	26	28	23	23	28	28



Mondlandung VC-64

Ein Raumschiff soll auf der Oberfläche des Mondes weich gelandet werden. Bei zu hartem Aufsetzen, bei zu starker seitlicher Abdrift oder beim Berühren der Felsen explodiert es. Wenn es zuweit aus dem Bildschirm verschwindet, hat es sich verirrt, was ebenfalls Minuspunkte bringt. Die Punktzahl errechnet sich nach der Schwierigkeit des Landeplatzes, nach der Entfernung des Landeplatzes vom Ausgangspunkt, nach dem noch vorhandenen Treibstoffvorrat und nach der Landegeschwindigkeit. Bei mehreren Spielen erscheint für jeden Teilnehmer im selben Durchgang die gleiche Landschaft, die vom Zufallsgenerator gezeichnet wird.

Die Steuerung erfolgt wahlweise mittels Joystick (in Control-port 1) rechts und links mit dem Steuerknüppel, die Bremsraketen werden mit der Feuertaste gezündet, oder über die Tastatur. Nach links mit ←, nach rechts mit 1. Gebremst wird mit der Spacetaste. Zubeachten ist dabei die Trägheit, d.h. das Raumschiff reagiert nicht sofort, sondern mit Verzögerung. Mit S kann der Countdown unterbrochen werden, mit der Spacetaste oder mit dem Feuerknopf wird er abgekürzt.

Die Schwierigkeitsgrade können in folgenden Zeilen verändert werden:

Zeile 2
T% = Treibstoffmenge

Zeile 5106
V = Fallgeschwindigkeit

Zeile 5107
RX = seitliche Abdrift



```
1 VC=53248:SI=54272:POKEVC+16,0:POKEVC+21,0:POKE53280,12:POKE53281,0:HI=-300
2 JO=56321:J3=255:TX=120
3 FORA=0TO7:POKEVC+A,0:NEXT:Q=VC+31:POKEVC+42,0
10 DATA0,126,0,1,255,128,7,255,224,15,255,240,24,66,24,16,66,8,48,66,12
20 DATA48,66,12,63,255,252,63,193,252,63,221,252,31,213,248,31,221,248
30 DATA15,221,240,7,193,224,1,255,128,2,126,64,4,24,32,8,60,16,16,0,8
40 DATA124,0,62
100 POKE2040,13:FORA=0TO62:READD:POKE832+A,D:NEXT
2000 PRINT"□":FORA=1984TO2023:POKEA,160:POKEA+54272,11:NEXT
2010 FORA=1TO20:B=INT(RND(1)*900+1000):POKEB,42:POKEB+54272,7:NEXT
2015 POKESI+24,15:POKESI+5,253:POKESI+6,16:POKESI+1,100:POKESI,100:POKESI+4,129
2020 POKEVC+23,1:POKEVC+29,1:POKEVC+21,1:FORA=0TO200STEP1.3:POKEVC,A/1.3
2025 POKESI,100:POKESI+1,207-A:POKEVC+1,A
2030 FORB=1TO20:NEXT:NEXT:FORA=1TO300:NEXT:PRINTTAB(14)"■■■■■MONDLANDUNG"
2040 FORA=1TO600:NEXT:PRINT"■■        EIN SPIEL VON DENNIS MERBACH"
2050 FORA=1TO600:NEXT:FORA=1TO7:PRINT"                                  ";
2060 NEXT:INPUT"■■          WIEVIELE SPIELER";SP:DIMSP$(SP),S%(SP),J$(SP)
2070 POKEVC+1,208:FORA=1TOSP:PRINT"■■■■■","  SPIELER";A
2080 INPUT"■■          NAME";SP$(A)
2085 IFLEN(SP$(A))>19THENPRINT,"ZU LANG":GOTO2080
2090 PRINT,"■■■JOYSTICK ODER ■K■EYBOARD?"
2095 GETJ$(A):IFJ$(A)=""THEN2095
2096 IFJ$(A)<>"J"ANDJ$(A)<>"K"THEN2095
3000 NEXT:Y=208:X=200/1.3:GOSUB50000:PRINT"□":POKE649,1
4500 A=RND(-TI):FORA=1984TO2023
4510 IFZE>2THENZE=0
4520 IFRND(1)<.4THENS=INT(RND(1)*5-2)
4530 IFS=0ANDRND(1)<.35THENZE=ZE+1
4540 Z=Z+S:IFZ<2THENS=2:GOTO4540
4550 IFZ>12THENS=-2
4560 Z1=Z:IFA=1984ORA=2023THENZ1=24
4570 IFA=1985ORA=2022THENZ1=15
4580 FORB=ATOA-40*Z1STEP-40:POKEB,160:POKEB+SI,11:NEXT:NEXT
4700 FORS=1TOSP:PRINT"■","  PILOT:  ";SP$(S):PRINT," COUNTDOWN LAEUFT"
4701 IFJ$(S)="K"THENJ1=1:J2=2:GOTO4710
4702 IFJ$(S)="J"THENJ1=8:J2=4
4710 FORZ=10TO0STEP-1:POKESI+5,36:POKESI+6,0
4711 PRINT"■■■■■"TAB(18);Z;"■ ":POKESI+4,0:POKESI+4,33
4715 IFPEEK(JO)=J3-16THENZ=1
4720 IFZ=0THENPOKESI+6,255
4730 GETA$:IFA$="S"THEN4900
4740 POKESI,30:POKESI+1,70:FORW=1TO900:NEXT:NEXT
4750 PRINT"■";:FORA=1TO5:PRINT"■■                                ":NEXT:GOTO5000
4900 POKESI+4,0:PRINT"■■■                              ":FORA=1TO1000:NEXT
4910 IFPEEK(JO)=J3-16THEN4930
4920 GETA$:IFA$=""THEN4910
4930 PRINT,"□ COUNTDOWN LAEUFT":POKESI+4,33:GOTO4740
5000 PX=0:TR=TX:RX=2:X=50:Y=30:POKEVC+27,15:POKEVC+29,0:T=0
5001 POKEVC+1,Y:POKEVC,X:POKEVC+23,0:POKEVC+21,7:POKEVC+31,0:POKE2040,13
5003 POKEVC+2,X:POKEVC+3,Y:POKEVC+4,X:POKEVC+5,Y:P=PEEK(Q):POKE214,24
5004 PRINT"□":POKEVC+16,0:POKESI+4,0
5005 POKE2041,14:POKE2042,15:FORA=896TO1022:POKEA,0:NEXT:POKESI+24,4
5006 POKE956,16:POKE1022,8:POKESI+1,20:POKESI+5,64:POKESI+6,255
5101 J=PEEK(JO):IFTR<=0THENTR=0:V=V+2:POKESI+24,0:GOTO5105
5102 IF(JAND16)=0THENT=T+.5+V/12:TR=TR-1:POKESI+4,129
5103 IF(JANDJ2)=0THENRX=RX-.8:TR=TR-1:POKESI+4,129
5104 IF(JANDJ1)=0THENRX=RX+.8:TR=TR-1:POKESI+4,129
5105 FORWA=1TO4:NEXT:P=PEEK(Q):IFP<>0THENPOKESI+4,0:GOTO10000
5106 V=Y/50+V/2-T/1.2:IFT>0THENT=T-.3
5107 RX=RX-SGN(RX)*ABS(RX)/5:X=X+RX:PX=PX+RX
5108 IFX<255ANDX>0THEN5113
5109 IFX>255ANDPEEK(VC+16)=7THEN6000
5110 IFX>255THENPOKEVC+16,7:X=0:GOTO5113
5111 IFX<0ANDPEEK(VC+16)=0THEN6000
5112 IFX<0THENX=255:POKEVC+16,0
5113 POKEVC,X:POKEVC+2,X:POKEVC+4,X:Y=Y+V:IFY<0ORY>255THEN7000
5114 POKEVC+1,Y:POKEVC+3,Y+1:POKEVC+5,Y+1
5115 PRINTCHR$(18)TAB(15)TR;"■■ □":TL=TR:IFJ=J3THENPOKESI+4,0
5116 GOTO5101
6000 PRINT"■","■SIE HABEN SICH VERIRRT":PX=-250:GOTO7020
7000 PRINT"■","■SIE HABEN DAS SCHWEREFELD":PRINT,"DES MONDES VERLASSEN UND"
7010 PRINT,"UND TREIBEN EWIG IM ALL":PX=-250
7020 PX=-250:POKESI+4,0::V1=SQR(V↑2+RX↑2):GOTO11110
10000 V1=SQR(V↑2+RX↑2):IFP=6THEN11000
10050 X=X-16:Y=Y-16:GOSUB50000:PRINT"■","ABGESTUERZT":POKEVC+21,8:X=X+16:Y=Y+16
10100 GOSUB20000:IFP1=0ORP2=0THENPU=1
10110 PX=(ABS(PX)/6-3*V1):IFPX<0ANDPU>1THENPX=0
10115 PX=INT(PU*PX)-60:IFPX>0THENPX=0
10120 GOTO11110
11000 IFV>1.5ORABS(RX)>.6THEN10050
11010 POKEVC+21,15:GOSUB20000:PX=INT(PU*(ABS(PX)/6+TR-3*V1))
11015 PRINT"■","GUT GELANDET"
11110 PRINT,"TREIBSTOFF";TR;"L":PRINT,"GESCWINDIGKEIT";LEFT$(STR$(V1),6);" KM/H"
11120 PRINT,"PUNKTE";PX
11130 FORA=1TO400:NEXT:WAITJO,16,J3
11140 PRINT"■";:FORA=1TO6:PRINT"■■                                ":NEXT
11200 S%(S)=S%(S)+PX:POKEVC+21,0:POKEVC+21,0:IFPX>HITHENHI=PX:HI$=SP$(S)
11210 NEXT:N%=N%+1:PRINT"□■":PRINT,N%;". RUNDE"
11220 FORA=1TOSP:PRINTTAB(5)"■";SP$(A):PRINT"□",,S%(A);"PUNKTE":NEXT
11230 PRINTTAB(6)"■■■HOECHSTPUNTZAHL";HI;HI$
11240 FORA=1TO400:NEXT:WAITJO,16,J3
11250 PRINT"□":GOTO4500
20000 POKE2043,14:PU=1:POKEVC+7,Y+1:P1=0:P2=0
20005 IFPEEK(VC+16)=7THENPOKEVC+16,15
20006 XA=X-6:IFXA<0THENPOKEVC+16,7:XA=255+XA
20007 IFXA>255THENXA=XA-255:POKEVC+16,15
20010 POKEVC+6,XA:A=PEEK(Q):FORA=1TO15:NEXT:IFP=6AND(PEEK(Q)AND8)=0THENPU=PU+.5
20015 IFXA>255THENXA=XA-255:POKEVC+16,15
20020 POKEVC+7,Y-7:A=PEEK(Q):FORA=1TO15:NEXT:IF(PEEK(Q)AND8)=8THENPU=PU+1:P1=1
20030 POKEVC+7,Y-15:A=PEEK(Q):FORA=1TO15:NEXT:IF(PEEK(Q)AND8)=8THENPU=PU+1
20040 POKEVC+7,Y-22:A=PEEK(Q):FORA=1TO15:NEXT:IF(PEEK(Q)AND8)=8THENPU=PU+1
20045 XA=X+26:IFXA>255THENPOKEVC+16,PEEK(VC+16)+8:XA=XA-255
20050 POKEVC+6,XA:A=PEEK(Q):FORA=1TO15:NEXT:IF(PEEK(Q)AND8)=8THENPU=PU+1
20060 POKEVC+7,Y-15:A=PEEK(Q):FORA=1TO15:NEXT:IF(PEEK(Q)AND8)=8THENPU=PU+1
20070 POKEVC+7,Y-7:A=PEEK(Q):FORA=1TO15:NEXT:IF(PEEK(Q)AND8)=8THENPU=PU+1:P2=1
20080 POKEVC+7,Y+1:A=PEEK(Q):FORA=1TO15:NEXT:IFP=6AND(PEEK(Q)AND8)=0THENPU=PU+.5
20085 XA=X+8:IFXA>255THENPOKEVC+16,PEEK(VC+16)+8:XA=XA-255
20090 POKEVC+6,XA:A=PEEK(Q):FORA=1TO15:NEXT:IFP=6AND(PEEK(Q)AND8)=0THENPU=PU+.5
20100 RETURN
50000 F1=0:F2=2:POKESI+5,5:POKESI+6,0:POKESI+24,15
50010 POKESI+7,100:POKESI+8,8:POKESI+12,10:POKESI+11,129:POKESI+13,16
50020 IFX<0THENPOKEVC+16,0:X=255
50030 POKEVC+1,Y:POKEVC,X:POKEVC+23,1:POKEVC+29,1
50040 FORB=1TO12:POKE2040,B:H=F1:F1=F2:F2=H
50045 POKESI+4,0:POKESI+1,6-B/2:POKESI+4,129:POKESI+24,15-B/3
50050 POKE53281,F1:NEXT
50060 POKEVC+21,0:POKE649,9:POKESI+4,0:POKESI+24,15:POKESI+11,0:RETURN

READY.
```

Wurm VC-64

Der Wurm soll Futter (Herzchen) fressen, muß aber das Gift (Kreuzchen) liegen lassen. Er darf nicht gegen die Wand laufen oder sich selbst anbeißen. Passiert dies, skelettiert er. Die Todesursache wird angezeigt. Steuerung über Joystick (Control-port 2) oder über die Tastatur mit den Buchstaben I, J, K, M. Der Wurm startet bei der ersten eingegebenen Bewegungsrichtung. Das Anzeigen der Todesursache u. s. w. kann abgekürzt werden durch Betätigen des Feuerknopfes oder einer beliebigen Taste während des Skelettierens. Bei der Punktberechnung wird die Länge des zurückgelegten Weges negativ berücksichtigt. Jeder Spieler hat 3 Würmer, wenn alles Futter gefressen ist, wird im nächsten Level die Zahl des Giftes und die Geschwindigkeit erhöht.

Dennis Merbach

```
0 REM    *******  "WURM-3"   *******
1 JO=56320:FB=12:FU=1:FW=7:FH=2:FK=0:Y=9:SI=54272:GOTO50000
2 TS=100:GS=30:FS=50:LW=5:A$="I":B$="M":C$="J":D$="K":DIMP(1000)
3 PRINTCHR$(8);"▒":POKE53281,FU:POKE53280,FB:W=3:L%=1:L=LW:T=TS:G=GS:F=FS
5 GOTO20000
100 FORA=0TO39:POKE1064+A,102:POKE55336+A,Y:POKE56256+A,Y:POKE1984+A,102:NEXT
110 FORA=0TO920STEP40:POKE1064+A,102:POKE1103+A,102:POKE55336+A,Y:POKE55375+A,Y
120 NEXT:T%=T
200 FORA=0TOL-1:P(A)=1690+A:POKE55962+A,FW:POKEP(A),160:NEXT:POKEP(0),81
210 FORA=1TOF
215 PF=INT(960*RND(1))+40:IFPEEK(1024+PF)<>32THEN215
220 POKE1024+PF,83:POKE55296+PF,FH:NEXT:FORA=1TOG
230 PG=INT(RND(1)*960)+40:IFPEEK(PG+1024)<>32THEN230
240 POKE1024+PG,43:POKE55296+PG,FK:NEXT
298 GETR$:IFR$<>""THEN298
299 PRINT"▒▒PUNKTE";S:PRINT"▒",,"LEVEL";L%,"WUERMER";W
300 IFJ$="J"THEN310
301 GETR$:IFR$=""THEN300
302 GOTO1000
310 IF(PEEK(JO)AND15)=15THEN310
315 POKE649,0:R=0:GOTO1010
999 GETR$
1000 IFR$=A$THENR=-40:GOTO1004
1001 IFR$=B$THENR=40:GOTO1004
1002 IFR$=C$THENR=-1
1003 IFR$=D$THENR=1
1004 FORA=LTO1STEP-1:P(A)=P(A-1):NEXT:P(0)=P(0)+R:IFPEEK(P(0))<>32THEN2000
1005 POKESI+P(0),FW:POKEP(0),81:POKEP(1),160:POKEP(L),32
1006 FORA=1TOT:NEXT:N=N+1:IFJ$="K"THEN999
1010 J=PEEK(JO):IFABS(R)<>40AND(JAND1)=0THENR=-40:GOTO1004
1011 IFABS(R)<>40AND(JAND2)=0THENR=40:GOTO1004
1012 IFABS(R)<>1AND(JAND4)=0THENR=-1:GOTO1004
1013 IFABS(R)<>1AND(JAND8)=0THENR=1
1014 GOTO1004
2000 P%=PEEK(P(0)):IFP%<>83THEN2020
2001 SP=SP+1:L=L+1:P(L)=0:S=S+10:PRINT"▒▒PUNKTE";S:IFSP=FTHENSP=0:GOTO3000
2002 IFT%>1THENT%=T%-4
2003 GOTO1005
2020 IFP%=102THENP$(1)="▒SICH DEN ⁄OPF EINGERANNT":P$(2)="DU \OERDER!":GOTO5000
2030 IFP%=43THENP$(1)="▒HAT |IFT GEFRESSEN":P$(2)=" OUERG":GOTO5000
2040 P$(1)="▒SICH SELBST ANGEKNABBERT":P$(2)="DU \OERDER!":GOTO5000
3000 POKEP(0),81:POKEP(1),160:POKE54272+P(0),FW
3001 FORA=1TO1000:NEXT
3005 POKE53281,13:PRINTCHR$(14);"▒▒▒▒▒▒▒▒▒▒   ♠IE HABEN DEN OURM GEFUETTERT!"
3010 FORA=1TO(N/3):S=S-1:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒","▒UNKTE:";S;"▒  ":NEXT:N=0
3015 FORA=1TO1000:NEXT
3020 L%=L%+1:SP=0:T=T/2:G=G+15:GOTO6010
4000 FORA=1TO2000:NEXT:POKE53281,0
4005 PRINTCHR$(14);"▒▒▒▒▒▒▒▒▒▒   ♠LLE OUERMER SIND TOT!"
4010 FORA=1TOINT(N/3):S=S-1:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒","▒UNKTE:";S;"▒  ":NEXT
4015 N=0:PRINT,"▒ IOECHSTPUNKTZAHL";HI
4020 POKE649,9:PRINT"▒▒▒▒ ⁄OCH EIN ♠PIEL ( Y⁄)?"
4025 GETN$:IFN$<>""THEN4025
4030 GETN$:IFPEEK(JO)<>127THENN$="J"
4031 IFN$=""THEN4030
4033 IFN$<>"N"THENPRINTCHR$(142):GOTO4040
4035 PRINTCHR$(9);CHR$(142);"▒":PRINT"VOR AUFNEHMEN RUNSTOP/RESTORE DRUECKEN":EN
```

```
D
4040 IFHI<STHENHI=S
4050 SP=0:S=0:GOTO3
5000 FORA=1TO800:NEXT:POKESI+P(0),0:POKEP(0),87
5010 FORA=1TOL:GETX$:IFSO=1THENNEXT:SO=0:GOTO6000
5015 IFPEEK(JO)<>127ORPEEK(203)<>64THENSO=1
5020 FORB=1TO100:NEXT:POKESI+P(A),0:POKEP(A),91:POKEP(0),87:NEXT
5030 FORA=1TO1000:NEXT:POKE53281,0:PRINTCHR$(14)"           ER OURM HAT ";
5040 PRINTP$(1);FORA=1TO1500:NEXT:PRINT" ":FORA=1TO200:PRINTP$(2);:FORB=1TO15
5050 NEXT:NEXT:FORA=1TO1000:NEXT
6000 W=W-1:IFW=0THEN4000
6010 POKE53281,FU:PRINTCHR$(142);" ":L=LW:SP=0:GOTO100
20000 POKE649,9:PRINTCHR$(14);"               ITTE WAEHLEN           "
20010 PRINT,"    1  PIELBEGINN"
20013 PRINT,"   ":PRINT,"  2  OECHSEL  OYSTICK/ EYBOARD"
20015 PRINT,"   ":PRINT,"  3  OERTE AENDERN"
20016 PRINT,"  ":PRINT,"  4   NDE"
20019 PRINT,"    ITTE  UMMER EINGEBEN"
20020 GETN$:IFPEEK(JO)<>127THEN20031
20021 IFN$=""THEN20020
20025 IFN$="1"ORN$="2"ORN$="3"ORN$="4"THENA=VAL(N$):GOTO20030
20026 GOTO20020
20030 ONAGOTO20031,20032,20040,4035
20031 PRINTCHR$(142);" ":L=LW:F=FS:G=GS:T=TS:GOTO100
20032 IFJ$="J"THENJ$="K":GOTO20000
20033 IFJ$="K"THENJ$="J":GOTO20000
20040 PRINT"               ITTE WAEHLEN           "
20050 PRINT,"     1   NZAHL DES  IFTS":PRINT,"  ":PRINT,"  2   NZAHL DES  UTTERS"
20060 PRINT,"  ":PRINT,"  3   ESCHWINDIGKEIT":PRINT,"  "
20070 PRINT,"  4   ICHTUNGSSTEUERUNG":PRINT,"  ":PRINT,"  5  OURMLAENGE"
20073 PRINT,"  "
20075 PRINT,"  ":PRINT,"  6  OERTE IN  RDNUNG"
20080 GETN$:IFN$=""THEN20080
20090 IFN$="1"ORN$="2"ORN$="3"ORN$="4"ORN$="5"ORN$="6"THEN20100
20091 GOTO20080
20100 A=VAL(N$):ONAGOTO20120,20140,20160,20180,20200,20000
20120 PRINT"        NZAHL DES  IFTS:"," LTER OERT:";GS:PRINT,,"/EUER OERT";
20130 INPUTGS:GOTO20040
20140 PRINT"        NZAHL DES  UTTERS:"," LTER OERT:";FS:PRINT,,"/EUER OERT";
20150 INPUTFS:GOTO20040
20160 PRINT"        ESCHWINDIGKEIT:"," LTER OERT:";TS:PRINT,,"/EUER OERT:";
20170 INPUTTS:GOTO20040
20180 PRINT"      ICHTUNGSSTEUERUNG:  LTE OERTE:":PRINT,"    UFWAERTS",A$
20181 PRINT,"   BWAERTS",B$:PRINT,"  LINKS",C$:PRINT,"  ECHTS",D$
20182 PRINT,"  /EUE OERTE:":PRINT,"    UFWAERTS",
20183 GETA$:IFA$=""THEN20183
20184 PRINTA$:PRINT,"   BWAERTS",
20185 GETB$:IFB$=""THEN20185
20186 PRINTB$:PRINT,"  LINKS",
20187 GETC$:IFC$=""THEN20187
20188 PRINTC$:PRINT,"  ECHTS",
20189 GETD$:IFD$=""THEN20189
20190 PRINTD$:FORA=1TO1000:NEXT:GOTO20040
20200 PRINT"        OURMLAENGE:"," LTER OERT";LW:PRINT,"/EUER OERT (MAX10)";
20210 INPUTL:GOTO20040
50000 PRINT"    ","    I    /-------------\    I"
50010 PRINT,"         WURM     I    I":PRINT,"    \-------------/    "
50015 B=-2.7:FORA=14TO9STEP-1:B=B+2.79
50020 PRINTTAB(A)" EIN";:FORC=1TOB:PRINT" ";:NEXT:PRINT"SPIEL":
50030 FORC=1TO200:NEXT:IFA<14THEN50050
50040 PRINT,"         VON":PRINT,"   DENNIS MERBACH    ":FORC=1TO2000:NEXT
50050 NEXT:FORA=1TO1000:NEXT:PRINTTAB(13)" GROSSARTIGES":FORA=1TO2000:NEXT
50060 PRINT"              JOYSTICK ODER KEYBOARD (J/K)"
50070 GETJ$:IFPEEK(JO)<>127THENJ$="J":GOTO2
50075 IFJ$=""THEN50070
50080 IFJ$="J"ORJ$="K"THEN2
50090 GOTO50070
READY.
```

Geisterjäger

Geisterjäger ist ein weiteres Spiel unseres freien Mitarbeiters Dennis Merbach für den VC-64. Es nutzt die interessante Sprite-Grafik-Fähigkeit des Gerätes, um Figuren über den Bildschirm zu bewegen.

In diesem Spiel jagt ein Geisterjäger die herabschwebenden Gespenster fort, was ihm allerdings nur kurzzeitig gelingt. Dann haben sie ihn wieder erreicht und versuchen ihn in den Burggraben zu stoßen und zu ertränken.
Nur eine schnelle Auffassungsgabe bewahrt den Geisterjäger vor diesem Unglück.
Mit den Tasten 1 und 2 oder per Joystick kann der Jäger gesteuert werden und mit der Space-Taste oder der Feuertaste kann er seinen Speer gegen die Geister schleudern.

```
1 PRINT"▯▯":VC=53248:SI=54272:POKEVC+21,0
20 DATA0,54,0,0,124,0,0,215,0,0,215,0,0,215,128,28,255,140,31,187,156,15,199
21 DATA252,3,255,248,1,255,224,1,255,128,1,255,128,3,255,128,3,251,128,3,255,0
22 DATA3,223,0,3,255,0,1,127,0,1,255,0,1,244,0,0,240,0,0
23 DATA0,60,0,0,126,0,0,153,0,0,153,0,0,255,0,0,102,0,0,60,0,0,24,0
24 DATA160,24,0,191,255,0,191,255,128,160,61,192,0,60,224,0,60,112,0,60,56
25 DATA0,60,24,0,126,0,0,102,0,0,231,0,14,195,112,15,195,240
26 DATA0,64,224,80,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,0,0,0
100 FORA=832TO959:READB:POKEA,B:NEXT
102 FORA=960TO1023STEP3:READB:POKEA,B:POKEA+1,0:POKEA+2,0:NEXT
110 POKE2040,15:POKE2041,14:FORA=2042TO2047:POKEA,13:NEXT
111 POKEVC+27,255:POKEVC+39,0:POKEVC+40,12:FORA=41TO46:POKEVC+A,1:NEXT:GOTO10000
113 FORA=4TO14STEP2:POKEVC+A,85:POKEVC+1+A,78:NEXT
120 MS(2)=248:FORA=3TO7:MS(A)=MS(A-1)-2↑A:D(A)=28/(255-(A*45-100)):NEXT
140 PRINT"▯▯":FORA=1TO12:PRINT,,,"  ▯┬──┬──┬──┬──":PRINT,,," ┐ ▯┬──┬──┬──┬──┬.┐":NEXT
150 FORA=1984TO2023:POKEA,160:POKESI+A,12:NEXT:PRINT"▯",,,"▯▮ ▮▮ ▮▮ ▮▮ ▮▮ "
160 PRINT,,,"▮▮▮▮▮  ▯▮▮▮▮   ▯▮▮▮▮▮▮     ▯▔▔▯▮▮▮▮▮▮     ":FORA=2013TO2015:POKEA,32
170 POKEA+SI,9:NEXT:POKE2012,223
300 T$(1)="     ┌──────────────────┐":T$(2)="     | GEISTERJAEGER  |"
305 T$(3)="     | EIN SPIEL VON  |":T$(5)="     └──────────────────┘"
306 T$(4)="     | DENNIS MERBACH |"
310 POKEVC+21,4:POKEVC+16,4
320 FORA=85TO30STEP-1:POKEVC+4,A:NEXT:POKEVC+27,0
330 FORA=30TO0STEP-1:POKEVC+4,A:NEXT:POKEVC+16,0:FORA=255TO20STEP-1:POKEVC+4,A
340 NEXT:FORA=1TO21:PRINT"▯▯▯▯▯":FORB=1TO5:PRINTRIGHT$(T$(B),A):NEXT
350 POKEVC+4,20+A*8:FORB=1TO80:NEXT:NEXT
360 POKEVC+16,248:POKEVC+21,255
400 FORA=3TO7:FORB=90TO30STEP-1:POKEVC+2*A,B:NEXT:POKEVC+27,MS(A)
410 FORB=30TO0STEP-1:POKEVC+2*A,B:NEXT:POKEVC+16,MS(A):D=78
420 FORB=255TO(A*45-100)STEP-1:POKEVC+2*A,B:D=D-D(A)
430 POKEVC+1+2*A,D:NEXT:NEXT
440 PRINT"  ▯J▯OYSTICK ODER ▯K▯EYBOARD"
450 GETA$:IFPEEK(56321)=239THENA$="J":GOTO470
460 IFA$<>"J"ANDA$<>"K"THEN450
470 POKEVC+1,221:POKEVC+3,221:FORA=0TO150:POKEVC,A:POKEVC+2,A:NEXT
480 FORA=188TO255:POKEVC+4,A:NEXT:POKEVC+16,4:FORA=0TO30:POKEVC+4,A:NEXT
500 L=4:IFA$="K"THENL=1
2000 PJ=150:SX=PJ:POKEVC+30,0:P3=50:P4=50:P5=50:P6=50:P7=50:TI$="000000":C=0
2010 POKEVC,PJ:POKEVC+1,221:POKEVC+2,PJ:POKEVC+3,221:POKEVC+27,255
2020 PRINT"▯▯":FORA=1TO15:PRINT"                                                 ";:NEXT
2500 J=PEEK(56321):IF(JAND8)=0THENPJ=PJ+6.6+C
2501 IF(JANDL)=0THENPJ=PJ-6.6-C
```

```
2502 IF(JAND16)=0ANDSW<=0ANDSY=221THENSW=1
2503 IFSW>0THENSY=SY-16:SW=SW-.08
2504 IFSW<=0THENSW=0:SY=221:SX=PJ
2506 GETA$:IFA$><"V"THEN2512
2510 GOTO2510
2512 IFPJ>236ORPJ<24THEN5000
2513 POKE53249,SY:POKE53278,0:POKE53248,SX:POKE53250,PJ:C=C+.015
2514 SC=PEEK(53278):IF(SCAND8)=8THENP3=40:SW=0
2515 P3=P3+1.6+C:POKE53255,P3:IFP3>=200THENA=3:GOTO6000
2516 IF(SCAND16)=16THENP4=40:SW=0
2517 P4=P4+1.3+C:POKE53257,P4:IFP4>=200THENA=4:GOTO6000
2518 IF(SCAND32)=32THENP5=40:SW=0
2519 P5=P5+2.0+C:POKE53259,P5:IFP5>=200THENA=5:GOTO6000
2520 IF(SCAND64)=64THENP6=40:SW=0
2521 P6=P6+1.6+C:POKE53261,P6:IFP6>=200THENA=6:GOTO6000
2522 IF(SCAND128)=128THENP7=40:SW=0
2523 P7=P7+1.2+C:POKE53263,P7:IFP7>=200THENA=7:GOTO6000
2524 PRINT"▒",MID$(TI$,4,1);":"RIGHT$(TI$,2):GOTO2500
5000 IFPJ<24THENPJ=24:GOTO2513
5010 T$=TI$:PRINT"▒▒▒▒     SIE HABEN SICH ERTRAENKT!"
5020 POKEVC+1,221:FORA=237TO255:POKEVC+2,A:POKEVC,A:NEXT
5030 FORA=221TO255:POKEVC+3,A:POKEVC+1,A:NEXT
5040 PU=VAL(MID$(T$,3,2))*60+VAL(RIGHT$(T$,2))
5050 PRINT,"▒▒▒SIE HABEN SICH":PRINT,PU"SEKUNDEN"
5055 PRINT"                GEHALTEN"
5060 PRINT"     ALTER REKORD:";HI;"SEKUNDEN":IFPU>HITHENHI=PU
5090 WAIT56321,16,127:FORA=3TO7:POKEVC+2*A,A*45-100:POKEVC+2*A+1,28:NEXT
5100 POKEVC,150:POKEVC+1,221:POKEVC+21,255:POKEVC+2,150:POKEVC+3,221:SY=221:SW=0
5110 FORA=1TO1500:NEXT:GOTO2000
6000 T$=TI$:P=PEEK(VC+A*2):POKEVC,PJ:POKEVC+1,221:FORB=202TO221:POKEVC+A*2+1,B
6010 NEXT:RG=-1:IFPJ>PTHENRG=1
6020 POKEVC+A*2,P
6030 IFP<>INT(PJ)THENP=P+RG:GOTO6020
6040 P=P+2:POKEVC+A*2,P:POKEVC+2,P:POKEVC,P
6050 IFP<254THEN6040
6060 PRINT"▒▒▒ DAS GESPENST HAT SIE ERTRAENKT!":GOTO5020
10000 POKEVC+21,15:POKEVC+23,15:POKEVC+29,15
10010 POKEVC,40:POKEVC+1,208:POKEVC+2,40:POKEVC+3,208:POKEVC+6,30:POKEVC+7,50
10020 POKEVC+16,4:POKEVC+4,20:POKEVC+5,50
10030 PRINTCHR$(14),"▒    |EISTERJAEGER"
10040 PRINT"▒▒▒▒  ¯IE |ESPENSTER MOECHTEN ♠IE FANGEN!"
10050 PRINT"  ×ERSUCHEN ♠IE, SIE DARAN ZU HINDERN,"
10060 PRINT"  INDEM ♠IE SIE MIT DEM ♠PEER TREFFEN."
10070 PRINT"  ♠IE KOENNEN SICH NACH LINKS UND NACH "
10090 PRINT"  RECHTS BEWEGEN, ENTWEDER MITTELS"
10100 PRINT"  '1' UND '2' ODER MITTELS 'OYSTICK,"
10110 PRINT"  UND SIE KOENNEN MIT DER '♠PACE'- "
10120 PRINT"  BZW MIT DER _EUERTASTE DEN ♠PEER":PRINT"  WERFEN."
10130 PRINT"▒       _ALLEN ♠IE NICHT IN DEN |RABEN!"
10140 PRINT,"  ▒_ERTIG?"
10150 GETA$:IFA$=""THEN10150
10160 PRINTCHR$(142);"▒":POKEVC+23,0:POKEVC+29,0:POKEVC+21,0:POKEVC,0:POKEVC+2,0
10170 GOTO113

READY.
```

VC-64-Sprites-Generator

Sprite-Grafik ist eine verhältnismäßig neue Form, um Zeichen und Figuren über den Bildschirm zu bewegen. Beim VC-64 lassen sich Sprites zusätzlich zum normalen Videobild auf dem Bildschirm darstellen.

Das Programm soll das Erstellen von Sprites erleichtern, indem es durch die Cursorsteuertasten und die Space-Taste in einer 24*21-Matrix die Konstruktion von Sprites erlaubt.
Nach Erstellung aller Sprites werden die entsprechenden Bytes in Datazeilen übernommen und das Hilfsprogramm gelöscht.

Einschränkend sei vermerkt, daß maximal 4 verschiedene Sprites erstellt werden können, da mehr als 4 Sprites nicht problemlos abgespeichert werden können. Es dürfte allerdings kein Problem sein, das Programm entsprechend abzuändern. Außerdem können die Sprites nicht in Multicolor erstellt werden.

Rolf Werdehausen

Erklärung des Programms:

Zeile 20	Abfrage der Spriteanzahl (max. 4)
Zeile 60	Abfrage der Speicherbereiche (11,13,14,15)
Zeile 70	Abfrage der Zeilennummer ab der die Datazeilen gespeichert werden sollen (größer als 1000)
Zeile 100	Ausgabe des Zeichenfeldes Klammeraffe: Löschen des Spritepunktes F1 übernimmt fertig gezeichnete Bytes in DATA-Zeilen
Zeile 200	Sprites zeichnen und berechnen des Sprite-Punktes
Zeile 400	Sprites übernehmen
Zeile 650	Unterprogramm Punkte setzen/löschen
Zeile 690	Löschprogramm

```
10 PRINT"■■ERSTELLEN VON SPRITES!
20 PRINT"■WIEVIEL SPRITES";:INPUTSZ$
30 IF VAL(SZ$)<=0ORVAL(SZ$)>4THEN10
40 SZ=INT(VAL(SZ$)):PRINT"■IN WELCHE SPEICHERPLAETZE SOLL GEPOKT   WERDEN (";
50 PRINT"FREI SIND: 11, 13,14 UND 15)?
60 PRINT:FORI=1TOSZ:PRINT"SPRITES NR"I"■";:INPUTSB:SB(I)=SB*64:NEXT
70 PRINT"■DATA ZEILEN NR.: (>1000)          ■■■■■■■■■";:INPUTDA$
80 IFVAL(DA$)<=1000ORVAL(DA$)>65000THENPRINT"■■";:GOTO70
90 DA=INT(VAL(DA$))
100 DIMBS(21,3):I=0
110 POKE646,14:POKE198,0:I=I+1:IFI>SZTHEN690
120 PRINT"■■■SPRITES NR:"I"■"
130 PRINT:FORZ=1TO21:PRINTTAB(11)ZTAB(15)"************************":NEXT
140 PRINT"■■■■■■■■■■ZEICHNEN
150 PRINT"■■CURSOR/SPACE
160 PRINT"■ ■■ LOESCHEN
170 PRINT"     ■■(@)
180 PRINT"■■■  ■ENDE
190 PRINT"     ■■(F1)■
200 SP=1:S=1:ZZ=1:BS=7
210 PO=1024+SP+14+(ZZ+2)*40:IFPEEK(PO)=160THENPE=1
220 ZE=170:FA=7:GOSUB650:GETT$
230 ZE=42:FA=14:GOSUB650:IFPE=1THENZE=160:FA=7:PE=0:GOSUB650
240 IFT$=""THEN210
250 REM SPRITES ZEICHNEN
270 IFT$="■"ORT$=" "THEN340
280 IFT$="■"THEN380
290 IFT$="■"THEN430
300 IFT$="■"THEN450
310 IFT$="@"THEN470
320 IFASC(T$)=133THEN490
330 GOTO210
```

```
340 IFT$=" "THENIFPEEK(PO)<>160THENZE=160:FA=7:GOSUB650:BS(ZZ,S)=BS(ZZ,S)+2↑BS
350 IFSP=24ANDZZ<21THENZZ=ZZ+1:S=1:SP=1:BS=7:GOTO210
360 IFSP=24THEN210
370 SP=SP+1:BS=BS-1:GOTO400
380 IFSP=1THEN210
390 SP=SP-1:BS=BS+1
400 IFBS=-1THENBS=7:S=S+1:GOTO210
410 IFBS=8THENBS=0:S=S-1
420 GOTO210
430 IFZZ<>1THENZZ=ZZ-1
440 GOTO210
450 IFZZ<>21THENZZ=ZZ+1
460 GOTO210
470 IFPEEK(PO)<>42THENZE=42:FA=14:GOSUB650:BS(ZZ,S)=BS(ZZ,S)-2↑BS
480 GOTO210
490 REM SPRITES UEBERNEHMEN
500 FZ=DA:DA=FZ+INT((SZ+10)/10)*10
510 DA$=STR$(DA)+"DATA"
520 POKE646,6:PRINT"⊐"STR$(FZ)+"FORJ=0TO62:READQ:POKE"SB(I)"+J,Q:NEXT
530 D1=0:FORZZ=1TO21:FORS=1TO3
540 IFLEN(DA$)>70THENDA=DA+10:PRINTLEFT$(DA$,LEN(DA$)-1):DA$=STR$(DA+10)+"DATA"
550 DA$=DA$+RIGHT$(STR$(BS(ZZ,S)),LEN(STR$(BS(ZZ,S)))-1)+","
560 NEXT:NEXT
570 IFDA$=STR$(DA+10)+"DATA"THEN540
580 PRINTLEFT$(DA$,LEN(DA$)-1):DA=DA+10
590 PRINT"90DIMBS(21,3),SB("SZ"):I="I":SZ="SZ":DA="DA+10
600 PRINT"100";:FORN=1TOSZ:PRINT"SB("N")="SB(N)"▸";:NEXT:PRINT"▮ "
610 PRINT"500FZ="FZ+1
620 PRINT"GOTO90"
630 GOTO730
640 END
650 POKEPO,ZE:POKE54272+PO,FA:RETURN
690 ZA=10:ZE=120
700 POKE646,6:PRINT"⊐590ZA="ZE+10":ZE="ZE+110"
710 FORI=ZATOZESTEP10:PRINTI:NEXT:PRINT"GOTO690"
720 IFI>700THENPRINT"TTPOKE646,14"
730 POKE631,19:FORI=1TO14:POKE631+I,13:NEXT:POKE198,15:END
```

Disketten-schutz für Apple II (Teil 2)

Nachdem im letzten Heft ein Programm beschrieben wurde, mit dem Daten in einem nicht standartisierten Code auf Diskette aufgezeichnet wurden, wird hier nun ein Verfahren vorgestellt, das nur befugten Personen die Benutzung der Diskette erlaubt.

Das Wesentliche bei einem solchen Schutzsystem ist, daß der Benutzer keine Möglichkeit besitzen darf, in das Operationssystem (hier Applesoft-Basic mit DOS) zu kommen. Dies wird durch das Programm DOS PROTECTOR (Abb. 1) gewährleistet. Alle Abbruchbefehle wie RESET, CTRL-C, Erzwingen einer Fehlermeldung und auch IRQ und NMI werden von einem mit diesem Programm geänderten DOS mit einem „Aufhängen“ des Systems quittiert.

Um dieses Verhalten zu erreichen, werden im DOS auf Diskette einige Bytes geändert. So wird zum Beispiel der Programmteil „Fehlermeldung ausgeben“ so modifiziert, daß er die gleiche Aktion wie die RESET-Taste ausführt, nämlich den Sprung in eine Endlosschleife. Außerdem wird ein CTRL-C-Filter eingebaut. Wirkung: siehe oben.

Um nur befugtem Bedienungspersonal einen Zugriff auf Diskette zu gewährleisten, ist die Eingabe eines Codewortes sinnvoll. Ein konstantes Codewort allerdings schränkt dessen Funktion ein, da die Gefahr des „Abguckens“ doch relativ groß ist. Daher verlangt das Programm SPERRE, das in Abb. 2 zu sehen ist die Errechnung des einzugebenden Codezeichens. Eine 5 Zeichen lange Zeichenfolge, in der genau zwei Zahlen zwischen 0 und 4 vorkommen, wird angezeigt. Diese zwei Zahlen müssen addiert werden und das Ergebnis „geshiftet“ werden. Das heißt, die erhaltene Zahl muß zusammen mit der SHIFT-TASTE gedrückt werden. Eine Ausnahme bildet die Zahl 0: wenn sie als Summe erscheint, muß die Leertaste gedrückt werden. Als benutzerspezifischer Wert steht in Zeile 999 als DATA-Wert der Name der Diskette, der im oberen Bildschirmteil erscheint. Bei einer falschen Eingabe reagiert das Programm mit ... Aber lassen sie sich überraschen.

Das Programm SPERRE ruft seinerseits das Programm HELLO1 auf. Dieses Programm stellt ein einfaches Menüprogramm dar, das sich als Auswahl für bestimmte Programme eignet. Auch hier erscheinen die benutzerabhängigen Werte in DATA-Zeilen. In Zeile 999 wird der Name der Diskette abgelegt. Ab Zeile 1000 sind dann die einzelnen Programmnamen gespeichert. Als Zusatz zum Programmnamen wird noch eine Ziffer abgerufen, die die Art des Programmaufrufs spezifiziert. 1 steht dabei für BRUN, 2 für RUN und 3 für EXEC.

Die Programe sollen nun etwas genauer untersucht werden.

Zuerst DOS PROTECTOR: Diese Routine liest bestimmte Sektoren der Diskette ein, ändert einige Bytes und schreibt dann die geänderten Sektoren wieder auf die Diskette zurück. In den Zeilen 70, 110, 140, 170, 200 und 220 wird der einzulesende Sektor definiert und mit einem GOSUB 1040 gelesen.

Was bewirken jetzt aber die einzelnen POKES? In den Zeilen 70-100 wird der Teil des DOS geändert, der spezifiziert, welche Werte in den Speicherstellen 3DO-3FF stehen. In Zeile 90 werden diese Werte so umgestellt, daß die Speicherstellen von 3EF bis 3F1 und von 3F5 bis 3FF alle den gleichen Wert enthalten, nämlich ein JMP ($03F2). Dieser Befehl bewirkt einen Sprung zu der Adresse, die in $3F2 abgelegt ist. Genau das passiert aber auch bei einem RESET. Die Zeile 80 ändert den Teil des DOS um, der die RESET-Adresse errechnet. Normalerweise wird nämlich in $3F2 und $3F3 die gleiche Adresse wie in $3D1 und $3D2 gespeichert. Anstelle dessen steht dort nun die absolute Adresse $3F8. Da dort ein JMP ($03F2) steht, führt ein Reset zur Endlosschleife.

Zeile 120 bewirkt, daß, wenn nach einem LOAD-Befehl, aber vor Rücksprung in das Basic CTRL-C gedrückt wird, dieser Tastendruck ignoriert wird. Er würde nämlich vom DOS nicht verarbeitet werden und somit auch nicht zur Endlosschleife führen. Zeile 180 ändert das Fehlermeldungsmodul, so daß ein Sprung in die Endlosschleife erfolgt.

Zeile 205 schließlich erstellt den CTRL-C-Filter und den Unterprogrammteil, der CTRL-C nach einem LOAD-Befehl unterdrückt. Die Zeilen 150 und 230 sorgen dafür, daß dieser CTRL-C-Filter auch aktiviert wird.

Die Zeilen 1000-1035 stellen ein Unterprogramm dar, das den IOB (I/O-Control-Block) lokalisiert. Dieser IOB wird in den Unterprogrammen in den Zeilen 1040-1060 und 1070-1090 benutzt, um die RWTS (Read or Write a Track and Sector) zu steuern.

Nun zur Analyse des Programms SPERRE. In Zeile 1 werden zunächst zwei Zufallsfunktionen definiert. Die Funktion R1 liefert den ASCII-Wert eines zufälligen Zeichens zwischen 0 und 4, generiert also eine Zufallszahl. Die Funktion R2 hingegen generiert ein zufälliges Zeichen zwischen § und /. Zeile 4 ist sehr interessant. Hier wird nämlich in RL und RH der momentane RESET-Vector gespei-

chert. Anstatt dessen wird dort der Wert $E199 gespeichert. An dieser Stelle steht die Applesoft-Fehlermeldung „Illegal Quantity Error". Wichtig ist an dieser Stelle der Befehl CALL -1169. Warum, wird an späterer Stelle erläutert. In Zeile 5 steht nun der Befehl ONERR GOTO 210.

Durch die Änderung des RESET-Vektors wird also bei RESET die oben genannte Fehlermeldung ausgegeben. Diese wird aber vom ONERR GOTO-Befehl abgefangen und führt zu einem Sprung nach Zeile 210. Damit wurde also der Befehl ON RESET GOTO 210 simuliert. Die Zeilen 10-40 führen zur Ausgabe des oberen Bildschirmteils. Wichtig sind wieder die Zeilen 50-70. Diese Zeilen entstanden aus der Notwendigkeit, die Zufallsfunktion RND(1) wirklich zufällig zu machen. Denn das ist sie keinesfalls. Dieses kann mit Hilfe eines kleinen Programms leicht bewiesen werden, das als HELLO-Programm auf Diskette abgelegt wird, und Zufallszahlen erzeugt. Nach dem Kaltstart des Systems werden dort immer die gleichen Zahlen erscheinen.

Daher warten die Zeilen 50-70 des Porgramms SPERRE auf einen Tastendruck. So lange dieser nicht erscheint werden „Dummy-" Zufallszahlen erzeugt.

Die Zeilen 95-180, die durch die FOR-NEXT Schleife in Zeile 90 insgesamt 5 mal durchlaufen werden, fordern den Benutzer zur Eingabe eines Codezeichens auf. Da diese Zeilen relativ komplex aufgebaut sind, würde eine Analyse hier zu weit führen und bleibt dem Leser selbst überlassen.

Die Zeilen 210-310 werden bei einer Fehleingabe durchlaufen. Das Ergebnis soll hier nicht verraten werden, sondern selbst ausprobiert werden.

In Zeile 500 wird wieder der alte RESET-Vektor zurückgeschrieben. Der CALL 1002 ist zur Reaktivierung des DOS unbedingt notwendig.

Der Befehl CALL -1169, der nach jeder Modifikation des RESET-Vektors auftaucht hat folgenden Sinn: Um unterscheiden zu können, ob der Rechner gerade einen Einschalt-Reset ausführt oder einen Lauf-Reset, wird in der Speicherstelle $3F4 ein bestimmter Wert abgelegt, der vom Autostart-Monitor-ROM abgefragt wird. Sollte dieser Wert vorhanden sein, so wird ein JMP ($03F2) ausgeführt, andernfalls ein Cold-Boot. Um diesen Cold-Boot zu verhindern wird der Befehl CALL -1169 verwendet, der diesen bestimmten Wert in $3F4 schreibt. Da dieser Wert nicht konstant ist, muß er nach jeder Modifikation des RESET-Vektors neu errechnet werden. Näheres dazu findet sich in (1)-(6).

Das Programm HELLO1 ist sehr einfach aufgebaut. Auch hier findet der ONEER GOTO-Befehl Verwendung. Er wird benutzt, um das Ende der Data-Zeilen zu erkennen. In Zeile 10-55 wird der Bildschirm erstellt. Die Zeilen 60-80 geben die in den DATA-Zeilen gespeicherten Programmnamen mit einem vorausgestellten Kennbuchstaben aus. Dabei ist anzumerken, daß eine Leere DATA-Zeile eine Leerzeile auf dem Bildschirm bewirkt (siehe Zeile 70). In Zeile 99 wird der ONEER GOTO-Befehl wieder gelöscht, und es wird die Eingabe des Kennbuchstabens vom aufzurufenden Programm verlangt. Nach dessen überprüfung (Zeile 120) wird der entsprechende Programmname auf dem Bildschirm mit blinkender Schrift gekennzeichnet (Zeilen 140-185) und das Programm aufgerufen (Zeilen 190-210).

Eine optimal geschützte Diskette würde also das Bootprogramm das Programm HELLO von der SYSTEM MASTER-Diskette enthalten. Dieses Programm würde dann das Programm SPERRE aufrufen. Von dort aus wird dann das Menüprogramm HELLO1 aktiviert.

Wichtig: Aus den im ersten Absatz geschilderten Bedingungen darf natürlich kein Programm mit dem END-Befehl aufhören. Dieser Befehl bewirkt nämlich genau das, was mit dem Schutzsystem verhindert werden soll: den Ausstieg in die Kommandoebene des BASIC. Der Befehl END sollte ersetzt werden durch CALL 1016. Damit wird ein Sprung in die Endlosschleife erreicht.

Der Autor möchte noch erwähnen, daß dieses System keineswegs absolut knacksicher ist. Es ist aber eine einfache Möglichkeit, nur befugten Personal Zugriff zu bestimmten Disketten zu gewähren. Das typische Vorgehen beim Erstellen einer geschützten Diskette ist wie folgt:

Zuerst, wie im DOS-Manual beschrieben, das Standard-Hello-Programm von der System-Master-Diskette so modifizieren, daß das Programm SPERRE aufgerufen wird, nachdem die Language-Card geladen wurde. Der verbleibende END-Befehl ist zu ersetzen durch CALL 1016. Mit diesem Programm als Boot-Programm die Disketten nun initialisieren. Jetzt die Programme SPERRE und HELLO1 auf die neu initialisierte Diskette kopieren. Dabei jeweils den in Zeile 999 abgelegten Diskettennamen ändern. Im Programm HELLO1 sind daraufhin in die DATA-Zeilen ab 1000 die wählbaren Programme einzutragen (Bitte nicht die Codeziffer für den Aufruftyp vergessen). Eine leere DATA-Zeile bewirkt dabei eine Leerzeile auf dem Bildschirm. Als letztes werden die Programme auf die Diskette kopiert.

Bei Basic-Programmen ist dabei jeder END-Befehl durch CALL 1016 zu ersetzen. Bei Maschinenprogrammen ist das Vorgehen wesentlich komplizierter. Diese Beschreibung würde den Rahmen dieses Artikels sprengen.

Damit ist die Diskette bootfertig. Um sie nun zu schützen, muß zuerst das Programm DOS PROTECTOR aufgerufen werden, mit dem das DOS modifiziert wird. Danach kann das Programm SPECIAL DOS aktiviert werden, mit dem die Daten im Spezial-Format aufgezeichnet werden. Damit ist die Diskette geschützt und fertig für die Verwendung.

Vorschläge und Anregungen zu diesen Programmen richten Sie bitte an die Redaktion.

Jörg Bliesener

```
10  TEXT : HOME :A$ = "                                     ": INVERSE : PRINT A$A$"        D
O S   P R O T E C T O R         "A$A$;: NORMAL
20  VTAB 12: HTAB 1: PRINT "DRIVE ? (1/2) ->";
30  GET A$: IF A$ <  > "1" AND A$ <  > "2" THEN 30
40 DR =  VAL (A$)
50  GOSUB 1000
60  VTAB 12: HTAB 1: PRINT "DOS PROTECTION RUNNING..."
70 T = 0:S = 13: GOSUB 1040
80  POKE 560,169: POKE 561,3: POKE 562,234: POKE 571,169: POKE 572,248: POKE 573,234
90  POKE 625,248: POKE 626,3: POKE 630,108: POKE 631,242: POKE 632,3: POKE 633,108: POKE 634,242
: POKE 635,3: POKE 636,108: POKE 637,242: POKE 638,3: POKE 639,242: POKE 640,3
100  GOSUB 1070
110 T = 1:S = 3: GOSUB 1040
120  POKE 589,76: POKE 590,105: POKE 591,186
130  GOSUB 1070
140 T = 0:S = 0: GOSUB 1040
150  POKE 575,32: POKE 576,208: POKE 577,8: POKE 720,169: POKE 721,111: POKE 722,133: POKE 723,5
6: POKE 724,169: POKE 725,186: POKE 726,133: POKE 727,57: POKE 728,96
160  GOSUB 1070
170 T = 1:S = 5: GOSUB 1040
180  POKE 725,108: POKE 726,242: POKE 727,3
190  GOSUB 1070
200 T = 0:S = 4: GOSUB 1040
205  FOR I = 617 TO 633: READ A: POKE I,A: NEXT
210  GOSUB 1070
220 T = 0:S = 1: GOSUB 1040
230  POKE 580,32: POKE 581,208: POKE 582,8: GOSUB 1070
240  VTAB 12: HTAB 1: PRINT "DOS PROTECTION COMPLETED."
250  END
1000  DATA  32,227,3,133,7,132,6,96,165,7,164,6,32,217,3,96
1010  FOR I = 768 TO 783: READ J: POKE I,J: NEXT
1020  CALL 768
1030 IOB =  PEEK (6) + 256 *  PEEK (7)
1035  RETURN
1040  POKE IOB + 2,DR: POKE IOB + 3,0: POKE IOB + 4,T: POKE IOB + 5,S: POKE IOB + 8,0: POKE IOB
+ 9,2: POKE IOB + 12,1: POKE IOB + 13,0
1050  CALL 776
1060  RETURN
1070  POKE IOB + 2,DR: POKE IOB + 3,0: POKE IOB + 4,T: POKE IOB + 5,S: POKE IOB + 8,0: POKE IOB
+ 9,2: POKE IOB + 12,2: POKE IOB + 13,0
1080  CALL 776
1090  RETURN
10000  DATA 173,16,192,108,96,157,32,27,253,201,131,208,3,76,16,164,96
```

Abb. 1: DOS PROTECTOR

```
1  DEF  FN R1(X) =  INT ( RND (1) * 5) + 48: DEF  FN R2(X) =  INT ( RND (1) * 48) + 64
4 RL =  PEEK (1010):RH =  PEEK (1011): POKE 1010,153: POKE 1011,225: CALL  - 1169
5  ONERR  GOTO 210
10  HOME : FOR I = 1 TO 40:A$ = A$ + " ": NEXT
30  READ N$
40  INVERSE : PRINT A$;A$; SPC( 20 -  LEN (N$) / 2);N$; SPC( 40 -  POS (0));A$;A$: NORMAL
50  PRINT : PRINT : VTAB 14: HTAB 15: PRINT "PRESS ANY KEY"
60  POKE  - 16368,0
70  IF  PEEK ( - 16384) < 128 THEN X =  RND (1): GOTO 70
80  POKE  - 16368,0: VTAB 6: CALL  - 958
90  FOR I = 1 TO 5: VTAB 12: HTAB 18
95 X(1) =  FN R1(3):X(2) =  FN R1(4):C$(1) =  CHR$ ( FN R2(3)):C$(2) =  CHR$ ( FN R2(12)):C$(3)
=  CHR$ ( FN R2(2))
100 P(1) =  INT ( RND (1) * 4) + 1
110 P(2) =  FN R1(34) - 47: IF P(2) <  = P(1) THEN 110
120 CX = 1:CC = 1
130  FOR J = 1 TO 5
140  IF J = P(CX) THEN  PRINT  CHR$ (X(CX));:CX = CX + 1
150  IF J <  > P(CX) THEN  PRINT C$(CC);:CC = CC + 1
160  NEXT
170 X$ =  CHR$ (X(1) + X(2) - 64)
180  VTAB 15: HTAB 3: PRINT "CODE CHARACTER ->";: POKE  - 16368,0: WAIT  - 16384,128: GET C$
190  IF C$ = X$ THEN  NEXT
200  IF I = 6 THEN 500
210  RESTORE : READ N$: FOR J = 778 TO 798: READ I: POKE J,I: NEXT
230  FOR J = 1 TO 4: READ I: POKE 776,I: READ I: POKE 777,I: CALL 778: NEXT
270  HOME : FLASH : PRINT A$A$A$ SPC( 17)"+++++"; SPC( 35);"+++++"; SPC( 35);"+++++"; SPC( 25)"+
+++++++++++++++++++++++++"; SPC( 15);"++++++++++++++++++++++++++"; SPC( 15);"+++++++++++++++++++++
++++"; SPC( 8);
280 S$ = "                 +++++                    ": FOR I = 1 TO 15: PRINT S$;: NEXT
290  PRINT A$;: FOR W = 1 TO 5000: NEXT
300  HOME : NORMAL : VTAB 12: PRINT "   E V E R Y T H I N G   I S   O V E R
310  POKE 1010,0: POKE 1011,3: CALL  - 1169: POKE 768,76: POKE 769,0: POKE 770,3: CALL 768
500  POKE 1010,RL: POKE 1011,RH: CALL  - 1169: CALL 1002: PRINT : PRINT  CHR$ (4)"RUN HELLO1"
999  DATA  "GAMES III"
1000  DATA  173,48,192,136,208,5,206,9,3,240,9,202,208,245,174,8,3,76,10
1005  DATA  3,96
1010  DATA 200,150,200,150,200,150,240,255
```

Abb. 2: SPERRE

```
10  HOME
20  FOR I = 1 TO 40:A$ = A$ + " ": NEXT
30  READ N$: INVERSE : PRINT A$"  DOS 3.3"; SPC( 29 -  LEN (N$));N$;"  ";A$;
40  FOR I = 1 TO 20: PRINT A$;: NEXT
50  VTAB 4: PRINT "WELCHES PROGRAMM ?"
55  PRINT : PRINT
60 I = 1: ONERR  GOTO 99
70  READ A$: IF A$ = "" THEN  PRINT : GOTO 70
73  READ C
75  PRINT  CHR$ (I + 64)")  "A$
80 I = I + 1: GOTO 70
99  POKE 216,0
100  VTAB 22: PRINT "KENNBUCHSTABE EINGEBEN..-";: HTAB 25
105 I = I - 1
110  GET A$
120  IF A$ < "A" OR A$ >  CHR$ (I + 64) THEN 110
130  HTAB 25: PRINT A$
135 B$ = A$
140  RESTORE : READ A$
150  FOR I = 1 TO  ASC (B$) - 64
160  READ A$
165  IF A$ = "" THEN E = E + 1: GOTO 160
167  READ C
170  NEXT
180  VTAB 5 + I + E: FLASH : HTAB 5: PRINT A$
185  NORMAL
190  IF C = 1 THEN  PRINT  CHR$ (4)"BRUN "A$
200  IF C = 2 THEN  PRINT  CHR$ (4)"RUN "A$
210  IF C = 3 THEN  PRINT  CHR$ (4)"EXEC "A$
999  DATA  GAMES III
1000  DATA  GAME NO. 1,2
1020  DATA  GAME NO. 2,2
1030  DATA  SUPER GAME NO. 1,1
1040  DATA  EXEC FILE NO. 1,3
```

Abb. 3: HELLO1

Trollhöhlen

Obwohl für den Apple geschrieben, paßt das Programm in jeden Rechner. Die Befehle HOME und VTAB bedeuten: 'Lösche Bildschirm' und 'Vertikal-Tabulator'.

```
50  HOME : VTAB 5
52  PRINT "HOEHLE NR. 1": PRINT
55  PRINT "TROLL: 'DU DARFST DIR SOVIELE KUPFER-         MUENZEN NEHMEN WIE DU WILLST.
   DAZU SCHENKE ICH DIR SCHLUESSEL          NR 1.'"
60  PRINT : INPUT "TROLL: 'ALSO WIEVIELE MUENZEN WILLST DU          NEHMEN?' ";A$
65 KM =  VAL (A$):BM = 0:S = 1
100  HOME : VTAB 5
102  PRINT "HOEHLE NR. 2": PRINT
110  PRINT "TROLL: 'DU BEKOMMST SOVIELE KUPFERMUEN-        ZEN WIE DU BRONZEMUENZEN HAST
    UND UMGEKEHRT.'"
120 KM = KM + BM:BM = KM
130  GOSUB 5000
200  HOME : VTAB 5
202  PRINT "HOEHLE NR. 3": PRINT
210  PRINT "TROLL: 'WENN DU WENIGER ALS 3 MUENZEN          HAST, DANN SCHMEISSE ICH DICH IN
    EINE BODENLOSE GRUBE'"
220  IF BM + KM < 3 THEN 300
230  GOSUB 5000
290  GOTO 400
300  GOSUB 5000
310  HOME
320  END
400  HOME : VTAB 5
402  PRINT "HOEHLE NR. 4": PRINT
410  PRINT "TROLL: 'ICH NEHME MIR ZWEI BRONZEMUEN-         ZEN VON DIR!'"
420 BM = BM - 2: IF BM < 0 THEN BM = 0
430  GOSUB 5000
500  HOME : VTAB 5
502  PRINT "HOEHLE NR. 5": PRINT
510  PRINT "TROLL: 'ICH JAGE DICH ZURUECK, WENN DU         MEHR ALS EINE BRONZEMUENZE BEI
    DIR HAST."
520  GOSUB 5000
530  IF BM > 1 THEN 400
600  HOME : VTAB 5
602  PRINT "HOEHLE NR. 6": PRINT
610  PRINT "TROLL: 'WENN DU KEINE BRONZEMUENZEN HAST       DANN MUSST DU DIE TUER ZU DER
    DEIN SCHLUESSEL PASST OEFFNEN!'"
620  GOSUB 5000
630  IF BM >  < 0 THEN 700
640  IF S < 5 THEN 800
650  HOME : VTAB 12
652  PRINT "HOEHLE NR. 7": PRINT
660  PRINT "ZAUBERER: 'WENN DU MEHR ALS ZWEI KUPFER-       MUENZEN HAST VERWANDELE ICH
       DICH IN EINEN WURM!'"
670  GOSUB 5000
680  HOME : VTAB 12: IF KM > 2 THEN  PRINT "DU BIST EIN WURM.": END
690  PRINT "DU HAST ES GESCHAFFT...": END
700  HOME : VTAB 5
702  PRINT "HOEHLE NR. 9": PRINT
710  PRINT "TROLL: 'ICH TAUSCHE DEINEN SCHLUESSEL          GEGEN DEN NAECHST HOEHEREN.
    WENN DU MIR SCHLUESSEL NR.5             GIBST, DANN FRESSE ICH DICH."
720  IF S < 5 THEN S = S + 1: GOSUB 5000: GOTO 900
730  GOSUB 5000
740  PRINT "TROLL: 'ICH HABE DICH VIERMAL GEWARNT!'"
750  END
800  HOME : VTAB 5
802  PRINT "HOEHLE NR. 8": PRINT
810  PRINT "DU BIST IN EINE HOEHLE VON MENSCHENFRES-SENDEN BANDWUERMERN GERATEN UND WIRST GERA
```

```
DE VERSPEIST.": END
900  HOME : VTAB 5
902  PRINT "HOEHLE NR. 10": PRINT
910  PRINT "TROLL: 'ICH TAUSCHE DEINE BRONZEMUENZEN        GEGEN KUPFERMUENZEN.'"
920 KM = KM + BM:BM = 0
930  GOSUB 5000
1000  HOME : VTAB 5
1002  PRINT "HOEHLE NR. 11": PRINT
1010  PRINT "TROLL: 'ICH GEBE DIR EINE BRONZEMUENZE.'"
1020 BM = BM + 1
1030  GOSUB 5000
1100  HOME : VTAB 5
1102  PRINT "HOEHLE NR. 12": PRINT
1110  PRINT "TROLL: 'ICH NEHME MIR ZWEI KUPFERMUEN-          ZEN.'"
1120 KM = KM - 2: IF KM < 0 THEN KM = 0
1130  GOSUB 5000
1200  HOME : VTAB 5
1202  PRINT "HOEHLE NR. 13": PRINT
1210  PRINT "TROLL: 'DU DARFST NACH OBEN, WENN DU          KEINE KUPFERMUENZEN MEHR HAST.
    ANDERNFALLS GEHT ES DURCH DEN          ZWEITEN AUSGANG ZURUECK!'"
1230  GOSUB 5000
1240  IF KM = 0 THEN 100
1250  GOTO 1000
4999  END
5000  PRINT : PRINT "KUPFERMUENZEN: ";KM
5010  PRINT "BRONZEMUENZEN: ";BM
5020  PRINT "SCHLUESSEL NR.: ";S
5030  PRINT : PRINT "DRUECKE EINE TASTE. ";: GET A$: RETURN
```

Computerhirn = Mastermind

Für den CBM stellt Homecomputer heute 2 Mastermind-Versionen vor. In der einen erhält der Spieler die Option, selbst den versteckten Code zu finden, oder dem Computer die Aufgabe zu stellen! In der anderen spielen zwei Spieler gleichzeitig gegeneinander und jeder kann das Ergebnis des Gegners mit auswerten, um die Lösung zu finden.

Mastermind 1

Nachdem man das Mastermindprogramm eingetastet hat, darf man nicht gleich mit RUN starten! Zuerst wird RUN 2000 eingegeben. Dabei wird erst einmal überprüft, ob man die Zeilen 0 bis 5 exakt eingegeben hat. Das ist nötig, da dort das Maschinenprogramm hingePOKEd wird. Wenn alles in Ordnung ist, dann kann man das Programm starten.
Wenn der Computer die Zahl raten soll, dann wendet er folgende Strategie an:
Zuerst nimmt er Zufallszahlen als Versuch, bis er alle Ziffern einmal probiert hat; höchstens jedoch drei mal. Daß er alle in der Lösung vorkommenden Ziffern einmal probiert hat, merkt er daran, daß die Summe von allen Schwarzen und Weißen gleich Drei ist. Dabei kommt keine Ziffer doppelt vor, weil Ziffernkombinationen mit Ziffern, die schon einmal vorkamen, am Anfang nicht so informativ sind. Für den Menschen ist es vielleicht einfacher, wenn er auch schon am Anfang Ziffern wiederholt verwendet, weil er dann nicht so viel denken muß. Den Computer stört das nicht.
Falls unter den ersten drei Versuchen keine Ziffer doppelt vorkommt, weiß der Computer nach drei Versuchen, wenn man ihm noch nicht drei Schwarze und Weiße gegeben hat, daß die noch nicht probierte Ziffer in der Lösung vorkommt.
Dann wird erst das Maschinenprogramm benutzt. Und zwar wendet es folgende Strategie an: Es zählt von 000 bis 999. Bei jeder Ziffernkombination geht es alle Versuche durch und berechnet bei jedem Versuch, wieviel Schwarze und Weiße es hätte erhalten müssen, wenn die angenommene Kombination die Lösung wäre.
Wenn es bei allen bisherigen Versuchen genausoviele Schwarze und Weiße berechnet, wie es auch bekommen hat, so verwendet es die laufende Ziffernkombination als nächsten Versuch, d.h., daß es als jeweils nächsten Versuch die kleinste Zahl nimmt, bei der die Annahme, daß es sich um die Lösung handelt, zu keinem Widerspruch zu den bisherigen Versuchen führt.
Die Versuche werden nicht in dimensionierten Variablen, sondern in den zweiten Kassettenbuffer abgelegt.

```
0 REM COPYRIGHT BY CORNELIUS GEPPERT
1 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
2 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
3 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
4 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
5 REMCOMPUTER
70 GOTO300
75 R%=10*RND(1):IFZ(R%)THEN75
77 Z(R%)=1:RETURN
80 POKE285,N:SYS1068
100 A%=PEEK(269):B%=PEEK(268):C%=PEEK(267)
120 IFPEEK(271)=0THENPRINT:PRINTTAB(7)"ERRARE HUMANUM EST":GOTO1250
240 N=N+5
250 IFETHEN1140
260 GOTO430
280 END
300 PRINT"❏ ▓▓ HOMECOMPUTER - M A S T E R - M I N D "
305 PRINT"▓▓▓  WOLLEN SIE DIE ZAHL RATEN? ";
310 GETB$:IFB$=""THEN310
312 GOSUB1400:A$=B$
320 PRINT:PRINT:IFA$="N"THENPRINT"  SOLL DER COMPUTER GEGEN":PRINT:PRINT"  SICH
SELBER SPIELEN? ";
330 IFA$="N"THENWAIT158,1:GETB$:GOSUB1400
338 N=0:POKE280,1
340 FORM=267TO270:POKEM,N:NEXT
342 FORW=1TO1E3:NEXT:PRINT"❏
345 IFA$="J"THENE=0:GOTO1020
347 IFB$="J"THENE=1:GOTO1020
```

```
360 GOSUB75:A%=R%:GOSUB75:B%=R%:GOSUB75:C%=R%
425 IFE=1THEN1140
427 IFN=0THENPRINT"VERSUCH"
430 PRINT:PRINTA%B%C%
440 PRINT:INPUT" WIEVIELE SCHWARZE UND WEISSE ";S,W:SU=SU+S+W
445 IFS+W>3THENPRINT"MAN MUESSTE MASTER-MIND SPIELEN KOENNEN!":GOTO440
447 POKE896+N,A%:POKE897+N,B%:POKE898+N,C%:POKE899+N,S:POKE900+N,W
450 GOSUB1300
480 IFSU<3ANDN<10THENN=N+5:GOTO360
490 IFS<3THEN80
500 PRINT:PRINT"ICH HABE NUR"N/5+1"VERSUCH"LEFT$("E",1ANDN>0)" GEBRAUCHT":GOTO12
50
1020 A=INT(10*RND(1)):B=INT(10*RND(1)):C=INT(10*RND(1))
1030 IFE=1THEN360
1040 INPUT"WIE LAUTET DER ERSTE VERSUCH "V:GOTO1110
1050 GOSUB1300
1070 IFE=0THENN=N+5
1075 IFSU<3ANDN<10ANDETHENN=N+5:GOTO360
1080 IFS<3ANDETHEN80
1085 IFETHEN500
1086 IFS<3THEN1090
1087 PRINT:PRINT"SIE HABEN NUR"N/5+1"VERSUCH"LEFT$("E",1ANDN>0)" GEBRAUCHT":GOTO
1250
1090 INPUT"WIE LAUTET DER NAECHSTE VERSUCH"V
1110 A%=V/100+1E-4
1120 B%=10*(V/100-INT(V/100)+1E-3)
1130 C%=10*(V/10-INT(V/10)+1E-3)
1140 S1=1:S2=S1:S3=S1:S4=S1:S5=S1:S6=S1:S=0:W=S
1150 IFA=A%THENS=S+1:S1=0:S2=S1
1160 IFB%=BTHENS=S+1:S3=0:S4=S3
1170 IFC%=CTHENS=S+1:S5=0:S6=S5
1180 IFA%=BANDS2ANDS3THENW=W+1:S2=0:S3=S2
1190 IFA%=CANDS2ANDS5THENW=W+1:S5=0
1200 IFA=B%ANDS1ANDS4THENW=W+1:S1=0:S4=S1
1210 IFB%=CANDS4ANDS5THENW=W+1
1220 IFA=C%ANDS1ANDS6THENW=W+1:S6=0
1230 IFC%=BANDS3ANDS6THENW=W+1
1235 POKE896+N,A%:POKE897+N,B%:POKE898+N,C%:POKE899+N,S:POKE900+N,W:SU=SU+S+W
1240 GOTO1050
1250 PRINT"NOCHMAL SPIELEN (LASSEN) ?
1260 WAIT158,1:GETB$:IFB$="J"THENRUN
1270 PRINT"SCHADE !"
1280 PRINT:PRINT" AUF WIEDERSEHEN
1290 END
1300 IFEANDN>0THENFORN1=N1+5TON1STEP5:GOTO1330
1305 INPUT:PRINTTAB(22);"ANZAHL DER
1310 PRINT" VERSUCH"SPC(9)"SCHWARZEN UND WEISSEN
1320 FORN1=896TON+N1STEP5
1330 PRINTPEEK(N1)PEEK(1+N1)PEEK(2+N1),,PEEK(3+N1),PEEK(4+N1):NEXT:N1=N1-5
1340 RETURN
1400 PRINTLEFT$("JA",2ANDB$="J")LEFT$("NEIN",4ANDB$="N"):RETURN
2000 FORI=0TO5:READA:IFPEEK(A)ORPEEK(A+4)THENPRINT"DIE ZEILE"I"STIMMT NICHT.":EN
D
2010 NEXT
5000 DATA1061,1132,1203,1274,1345,1359
5010 FORI=1062TO1359:READA:POKEI,A:NEXT
5020 PRINT"JETZT KOENNEN DIE ZEILEN AB 2000
5030 PRINT"GELOESCHT WERDEN.
6000 DATA129,4,1,0,143,34,141,15,1,174,41,4,169,1,160,6,153,15,1,136,208,250,140
6010 DATA22,1,140,23,1,189,128,3,77,13,1,208,9,238,22,1,140,16,1,140,17,1,189,12
9
6020 DATA3,77,12,1,208,9,238,22,1,140,18,1,140,19,1,189,130,3,77,11,1,208,9,238
6030 DATA22,1,140,20,1,140,21,1,189,131,3,77,22,1,240,11,76,26,5,0,80,5,2,0,143
6040 DATA34,66,189,128,3,77,12,1,208,17,173,17,1,45,18,1,240,9,238,23,1,140,17,1
6050 DATA140,18,1,189,128,3,77,11,1,208,14,173,17,1,45,20,1,240,6,238,23,1,140,2
0
6060 DATA1,189,129,3,77,13,1,208,17,173,16,1,45,19,1,240,9,238,23,1,140,16,1,140
6070 DATA19,1,189,129,3,77,11,1,208,11,173,19,1,45,20,1,240,3,238,23,1,189,130,3
6080 DATA77,13,1,208,14,173,16,1,45,21,1,240,6,238,23,1,140,21,1,189,130,3,77,12
6090 DATA1,208,11,173,18,1,45,21,1,240,3,238,23,1,189,132,3,77,23,1,208,6,236,29
6100 DATA1,208,46,96,173,11,1,201,9,240,6,238,11,1,76,47,4,205,12,1,240,9,238,12
6110 DATA1,152,141,11,1,240,239,205,13,1,240,9,238,13,1,152,141,12,1,240,237,140
6120 DATA15,1,96,232,232,232,232,232,76,50,4,0
```

Mastermind 2

Bei dieser Version spielen zwei Spieler gegeneinander und versuchen zur gleichen Zeit die versteckte Zeichenkombination zu erraten, wobei jeder die Versuche des Gegners und die eigenen in seine Stategie zur Lösungsfindung mit einbeziehen kann.
Die Spielanleitung am Anfang erklärt, wie das Spiel bedient wird. Hat man seinen Versuch auf dem Bildschirm mitttels Zahlen und den 'Größer-als- Kleiner-als-Tasten' plaziert, so schließt man ihn mit der Spacetaste ab.

```
1 H=2:GOSUB1000
5 INPUT"WIEVIEL ZEICHEN ( 3 - 10 )";WZ:WZ=INT(WZ):IFWZ>10ORWZ<3THEN5
10 DIMTR$(2,12,5),RP(2,12),RZ(2,12),CZ$(5),P$(14)
20 FORI=1TO5:CZ(I)=INT(RND(1)*WZ):NEXT
21 FORI=1TO5:IFCZ(I)=0THENCZ$(I)="#"
22 IFCZ(I)=1THENCZ$(I)="♥"
23 IFCZ(I)=2THENCZ$(I)="♣"
24 IFCZ(I)=3THENCZ$(I)="*"
25 IFCZ(I)=4THENCZ$(I)="♠"
26 IFCZ(I)=5THENCZ$(I)="♦"
27 IFCZ(I)=6THENCZ$(I)="▛"
28 IFCZ(I)=7THENCZ$(I)="▚"
29 IFCZ(I)=8THENCZ$(I)="$"
30 IFCZ(I)=9THENCZ$(I)="▒"
31 NEXT
32 FORI=1TO14:READP$(I):NEXT
35 PRINT"□"
50 F=0:IFH=1THENH=2:GOTO70
60 IFH=2THENH=1
62 DATA"     ","      ","0 = #","1 = ♥","2 = ♣","3 = *","4 = ♠","5 = ♦","6 = ▛"
63 DATA"7 = ▚","8 = $","9 = ▒"
64 DATA"     ","      "
70 PRINT"▒▒     ** MASTERMIND **     |        "
75 IFV<1THENFORI=1TO14:PRINT"▒                              |    ";P$(I);"   ":NEXT
80 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒      ▒ ▬ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▬ "
82 PRINT"     |<▒▒        RETURN        |>▒ "
84 PRINT"     ▒▀▀▄▛ ▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▒▀▀▄▛ "
95 PRINT"▒▒"
100 FORI=1TO12:FORJ=1TO5
110 PRINT"▒▒|";LEFT$(TR$(H,I,J),1);:IFTR$(H,I,J)=""THENPRINT"▒.";
115 NEXT
120 PRINT"▒    ";:IFRP(H,I)=0THENPRINT"▒      ";:LF=5
121 IFRP(H,I)=1THENPRINT"▒●     ";:LF=4
122 IFRP(H,I)=2THENPRINT"▒●●    ";:LF=3
123 IFRP(H,I)=3THENPRINT"▒●●●   ";:LF=2
124 IFRP(H,I)=4THENPRINT"▒●●●● ";:LF=1
125 IFRP(H,I)=5THENPRINT"▒●●●●●";
126 FORI1=1TOLF:PRINT"|";:NEXT
130 IFRZ(H,I)=1THENPRINT"▒o";
131 IFRZ(H,I)=2THENPRINT"▒oo";
132 IFRZ(H,I)=3THENPRINT"▒ooo";
133 IFRZ(H,I)=4THENPRINT"▒oooo";
134 IFRZ(H,I)=5THENPRINT"▒ooooo";
140 PRINT
150 NEXT:IFF=1THEN50
200 F=1:L=L+1:V=INT(L/2.000001)+1
210 PRINT"▒▒SPIELER";H;"|            "
215 FORI2=0TO4:POKE32810+V*40+I2*3,191:NEXT
216 GOSUB600
221 GOSUB501
225 FORI=1TO5:CZ$(I)=LEFT$(CZ$(I),1):NEXT
230 FORI=1TO5:IFTR$(H,V,I)=CZ$(I)THENRP(H,V)=RP(H,V)+1:GOSUB900
235 NEXT
240 FORI=1TO5:FORJ=1TO5:IFTR$(H,V,I)=CZ$(J)THENRZ(H,V)=RZ(H,V)+1:GOSUB910
245 NEXTJ,I
250 IFRP(H,V)=5THEN300
260 IFV=12ANDH=2THEN400
270 GOTO50
300 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒SPIELER";H;"| GEWONNEN IM";V;"|. VERSUCH     ▒▒▒▒"
310 GOTO430
400 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
401 PRINT"▒ALLE SPIELER SIND NACH 12 VERSUCHEN▒      ▒GESCHEITERT!";
402 PRINT"▒                    "
```

PET-Bytes

```
405 PRINT"▩
410 PRINT"▩DER ZAHLENCODE LAUTETE:   ";:FORI=1TO5:PRINTLEFT$(CZ$(I),1);" ";:NEXT:
PRINT
420 PRINT"▩                                         "
430 INPUT"▩NEUES SPIEL";JN$:IFLEFT$(JN$,1)="J"THENRUN
440 PRINT"▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩OKAY..............................."
445 POKE158,0:WAIT158,1:POKE158,0
450 END
501 FORI=1TO5:IFTR$(H,V,I)="0"THENTR$(H,V,I)="#"
502 IFCZ(I)=1THENCZ$(I)="♥"
503 IFCZ(I)=2THENCZ$(I)="♣"
504 IFCZ(I)=3THENCZ$(I)="✱"
505 IFCZ(I)=4THENCZ$(I)="♠"
506 IFCZ(I)=5THENCZ$(I)="♦"
507 IFCZ(I)=6THENCZ$(I)="▛"
508 IFCZ(I)=7THENCZ$(I)="▚"
509 IFCZ(I)=8THENCZ$(I)="$"
510 IFCZ(I)=9THENCZ$(I)="▒"
511 NEXT
512 RETURN
600 I3=32810+V*40:I4=I3-1:POKEI4,158:POKE158,0
610 GETA$:IFA$=""ANDPEEK(I4)=191THENPOKEI3,63
612 IFA$=""ANDPEEK(I4)=63THEN610
614 IFA$="<"ANDI4>I3-1THENI4=I4-3:POKEI4,158:POKEI4+3,160
616 IFA$=">"ANDI4<I3+11THENI4=I4+3:POKEI4,158:POKEI4-3,160
618 IFA$=" "THENPRINT"▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩▩                                        ":GOT
O700
620 IFA$="0"THENPOKEI4+1,163
625 IFA$="1"THENPOKEI4+1,211
630 IFA$="2"THENPOKEI4+1,193
635 IFA$="3"THENPOKEI4+1,170
640 IFA$="4"THENPOKEI4+1,216
645 IFA$="5"THENPOKEI4+1,218
650 IFA$="6"THENPOKEI4+1,233
655 IFA$="7"THENPOKEI4+1,255
660 IFA$="8"THENPOKEI4+1,164
665 IFA$="9"THENPOKEI4+1,230
670 GOTO610
700 I6=0:FORI5=I3TO(I3+12)STEP3:I6=I6+1
701 IFPEEK(I5)=211THENTR$(H,V,I6)="♥"
702 IFPEEK(I5)=193THENTR$(H,V,I6)="♣"
703 IFPEEK(I5)=170THENTR$(H,V,I6)="✱"
704 IFPEEK(I5)=216THENTR$(H,V,I6)="♠"
705 IFPEEK(I5)=218THENTR$(H,V,I6)="♦"
706 IFPEEK(I5)=233THENTR$(H,V,I6)="▛"
707 IFPEEK(I5)=255THENTR$(H,V,I6)="▚"
708 IFPEEK(I5)=164THENTR$(H,V,I6)="$"
709 IFPEEK(I5)=230THENTR$(H,V,I6)="▒"
710 IFPEEK(I5)=163THENTR$(H,V,I6)="#"
712 NEXT
720 FORI5=I3-1TOI3+11STEP3:POKEI5,160:NEXT
730 RETURN
900 TR$(H,V,I)=TR$(H,V,I)+"●":CZ$(I)=CZ$(I)+"o":RETURN
910 TR$(H,V,I)=TR$(H,V,I)+"●":CZ$(J)=CZ$(J)+"o":RETURN
1000 PRINT"▩▩▩▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
1010 PRINT"▩▩ MASTERMIND #2 "
1040 PRINT"▩▩FINDEN SIE DEN VERSTECKTEN ZEICHENCODE."
1050 PRINT"MIT DEN TASTEN ▩ < ▩ & ▩ > ▩ HABEN SIE"
1055 PRINT"DIE MOEGLICHKEIT DEN PLATZ ZU BESTIMMEN,"
1060 PRINT"▩AN DEN SIE EIN BESTIMMTES ZEICHEN SET-"
1065 PRINT"ZEN MOECHTEN ( EINGABE EINER ZAHL! )."
1070 PRINT"MIT DER LEERTASTE BEENDEN SIE IHREN VER-"
1075 PRINT"▩SUCH."
1080 PRINT"DIE ▩●▩-ZEICHEN ZEIGEN AN: CODEZEICHEN ISTAN DER RICHIGEN STELLE!"
1085 PRINT"DIE ▩o▩-ZEICHEN ZEIGEN AN: CODEZEICHEN ISTRICHTIG ABER AN DER FALSCHE
N";
1090 PRINT" STELLE!"
1100 RETURN
```

Biete an Software

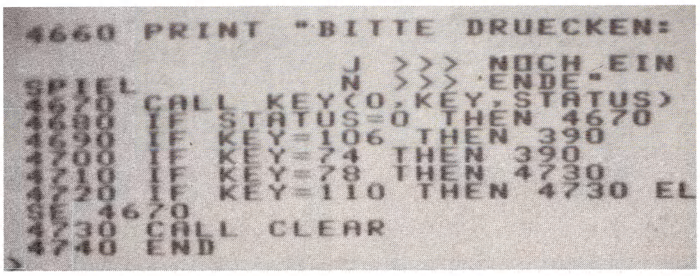
Biete an Hardware

Suche Software

Verschiedenes

Kontakte

Fortsetzung von Seite 7



Der VC-64 Videochip Teil 1

In diesem Heft beginnt eine Serie, in der dem Computer etwas näher in die Bytes geschaut werden soll.

Die maximale Anzahl der Zeilen im Tastaturpuffer läßt sich durch POKE 649, N bestimmen, wobei N eine Integerzahl zwischen 0 und 255 ist. Tippt man noch mehr Tastendrücke ein bevor die nächste GET Anweisung kommt, so werden die weiteren Tastendrücke ignoriert. Es ist jedoch nicht empfehlenswert, in 649 eine größere Zahl als 10 zu poken, da der 11te abgespeicherte Tastendruck den Computer aussteigen läßt.

Man kann das Basic umändern und sich sozusagen sein eigenes Basic schreiben. Dazu muß man erst das Basic aus dem ROM in den RAM-Bereich gleicher Adresse kopieren. Das macht man mit:

FORI = 40960 to 49151: POKEI, PEEK (I): NEXT.

Jetzt muß man dem Computer noch sagen, daß das Basic im RAM steht. Das tut man mit POKE 1,54. Zurückschalten kann man mit POKE 1,55.

Nun kann man nachsehen, an welcher Stelle die einzelnen Befehlsworte stehen; z. B. indem man die STOP-Taste drückt, wenn das gesuchte Wort kommt und dann die Variable I drückt. Dann PEEKed man die Bytes in der Nähe von I ab, um den genauen Wert zu bekommen. Per POKE kann man jetzt die Befehlsworte umändern. Man muß aber darauf achten, daß man bei dem letzten Buchstaben oder Zeichen das höchstwertige Bit setzt, d. h. zu dem ASCII-Code 128 addiert, und bei den anderen nicht, da sich sonst die Interpretercodes der darauffolgenden Befehle und Funktionen ändern. So kann man z. B. mit POKE 41118, ASC ("U") den Befehl END in UND umändern. Dann kann man schreiben: END = 7 : PRINTEND und erhält die 7. Dagegen führt die Anweisung END zu einem ?SYNTAX ERROR. Am Programmende kann man UND statt END schreiben.

Die Befehlsworte stehen ab 41118 bis 41372 in der Reihenfolge ihres Interpretercodes. Der erste Befehl, END, hat den Interpretercode 128. Das Ende eines Befehlswortes ist dadurch gekennzeichnet, daß das höchstwertige Bit gesetzt ist.

Wenn man die Befehlsworte auslisten will, muß man das höchstwertige Bit löschen, da sonst statt des letzten Buchstabens des Befehlswortes das Graphikzeichen erscheint, das man erhält, wenn man den letzten Buchstaben mit gedrückter Shift-Taste tippt. Hier ein Programm dazu:

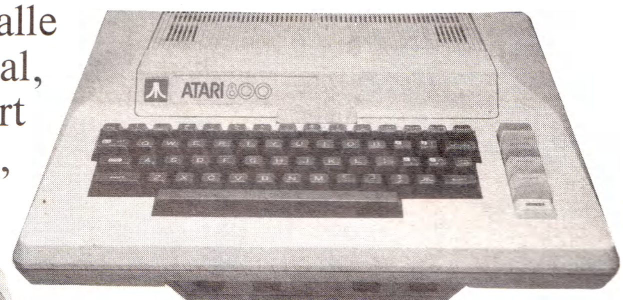
```
10 FOR I = 41118 TO 41372
20 P = PEEK(I):
PRINTCHR$ (127 AND
P); IF P > 127 THEN
PRINT
30 NEXT
```

Im nächsten Heft werden wir uns dann näher mit dem VC-64-Zeichensatz und seine Möglichkeiten befassen.

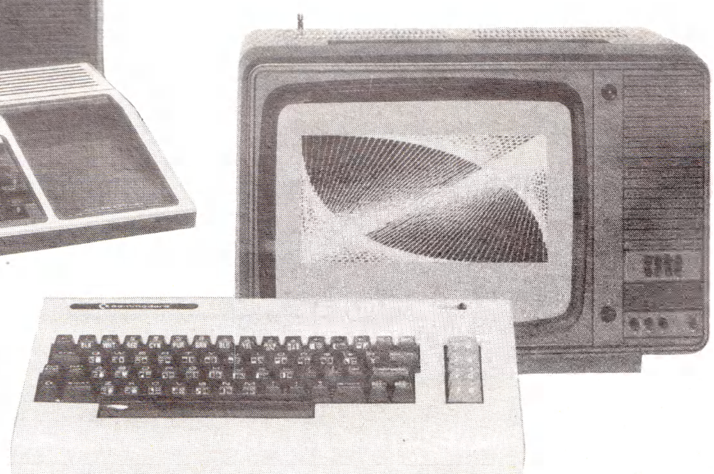
250 Computer zu gewinnen.

Jeder Abonnent, der einen Abonnenten für Homecomputer wirbt, erhält als kleines Dankeschön ”Das Microcomputer-Taschenbuch” (Erscheinungstermin Juni 1983) mit viel Wissenswertem über die populärsten Microcomputer.
Außerdem verteilen wir noch 250 Computer an diejenigen Leser, die Homecomputer die meisten Abonnenten zuführen.

Die Gewinnverteilung erfolgt nach einem für alle Leser sehr interessanten Schlüssel. Jedesmal, wenn bei uns 1000 neue Abonnenten registriert wurden, dann erhalten diejenigen 10 Leser, die zu diesem Zeitpunkt die meisten Abonnenten geworben haben – also in der Wettbewerbsliste vorne liegen – einen Microcomputer ihrer Wahl und zwar der Erste dieser Wettbewerbsliste einen Microcomputer im Werte von 2500.– DM und die 9 folgenden jeder einen im Werte von 1000,– DM.

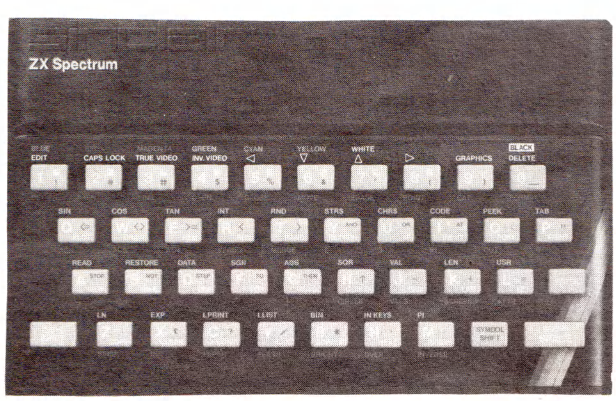






Die 10 Gewinner werden aus der Wettbewerbsliste gestrichen (sie können allerdings von neuem beginnen und wieder gewinnen), und die ihnen Folgenden rükken nach.

Sind wieder 1000 Abonnenten erreicht, geht es weiter wie oben beschrieben.



Teilnahmeberechtigt ist jeder, der Homecomputer regelmäßig im Abonnement bezieht:

Die Gewinner werden sofort benachrichtigt und in der jeweils folgenen Ausgabe von Homecomputer bekanntgegeben.

Leser-Software-Service

K=Kassette D=Diskette

TI-99/4A K 10,- DM
Panzerkrieg
TI-Ufo
Killersatellit
Chiffrier/dechiffrier-PGM
Rangliste

VC-20 K 14,- DM D 20,- DM
Invaders (3PGME)
Editor
Assembler (neu)
Labyrinth
Car-Crash

ZX-81 K 8,- DM
Ritter
Missile ZX-Command

VC-64 K 8,- DM D 15,- DM
Mondlandung
Wurm
Geisterjäger
VC-64-Sprite-Generator

Apple II D 12,- DM
Diskschutz
Trollhöhlen

Pet-Bytes K 8,- DM
Mastermind 1
Mastermind 2

Programme auf Diskette und Kassette

Nicht jeder Leser hat die Zeit und die Geduld, alle Programme, die er gerne hätte, in sein Gerät einzutasten.
Wir haben uns deshalb überlegt, daß es das einfachste ist, wenn wir die Programme, die wir ohnehin auf Diskette oder Kassette vorliegen haben, diesem Leserkreis zum Selbstkostenpreis (Datenträger, Kopierzeitaufwand, Porto, Verpackung und Mehrwertsteuer) anbieten.
Es genügt jeweils die Angabe eines Stichwortes, z. B.: VC-20-Disk Heft 5 oder Sinclair-Kassette Heft 5.

Bestellungen Inland:

Gegen Einsendung eines Schecks oder Vorauszahlung auf unser Konto bei der
Kreissparkasse Eschwege, Bankleitzahl 522 500 30 Nummer 45 22 934 senden wir Ihnen die gewünschten Programme schnellstmöglich zu.

Bestellungen Ausland:

Nur Vorauskasse Schein (Kassette 10,– DM, Diskette 20,– DM). Keine Schecks oder Überweisungen!

Lieferung noch nicht erhalten?

Bei Überweisung auf unser Konto kann es bis zu 2 Wochen dauern, bis wir Ihre Bestellung in Händen haben
Oft passiert es, daß auf der Überweisungsdurchschrift weder Name, noch Ort, noch Art der Bestellung zu erkennen sind. Schreiben Sie uns! (Anrufe kosten viel Geld und bringen, weil dann Schriftvergleiche nicht möglich sind, kein Ergebnis!)

Korrekturen

Heft 5/83

Spät, aber nicht zu spät fiel auf, daß das Listing vom **Assembler** in diesem Heft die „Generalproben-Version" war.

Später wurden noch 3 Zeilen geändert

1070 IF AI <6 AND L >1 THEN . . .
4135 AI = 0: (Rest wie abgedruckt)
10000 DATA (der Wert 170 muß durch 176 ersetzt werden)

Heft 4/83

Das Listing des Programmes **Asteroiden** hatte folgenden Fehler:

930 IF KEY=74 THEN 170

Auftrag für Gelegenheitsanzeigen in Homecomputer

An Homecomputer
Gartenstraße 47
3440 Eschwege

Name und Adresse ____________________

____________________ Abo.-Nr. ________

____________________ ________
Unterschrift Ich zahle sofort nach Rechnungserhalt. Datum

Unter der Rubrik „Kleinanzeigen" veröffentlichen wir Gelegenheitsanzeigen für Verkaufsangebote, Kauf- und Tauschgesuche, Kontaktaufnahme bzw. Erfahrungsaustausch usw.

Preise für „Kleinanzeigen": Private Gelegenheitsanzeige je Druckzeile 5,– DM inkl. MwSt. Chiffregebühr je Anzeige 10,– DM.

Gewerbliche Gelegenheitsanzeige je Druckzeile 7,– DM inkl. MwSt. (dürfen nicht unter Chiffre erscheinen).

Ich wünsche folgenden Text zu veröffentlichen:

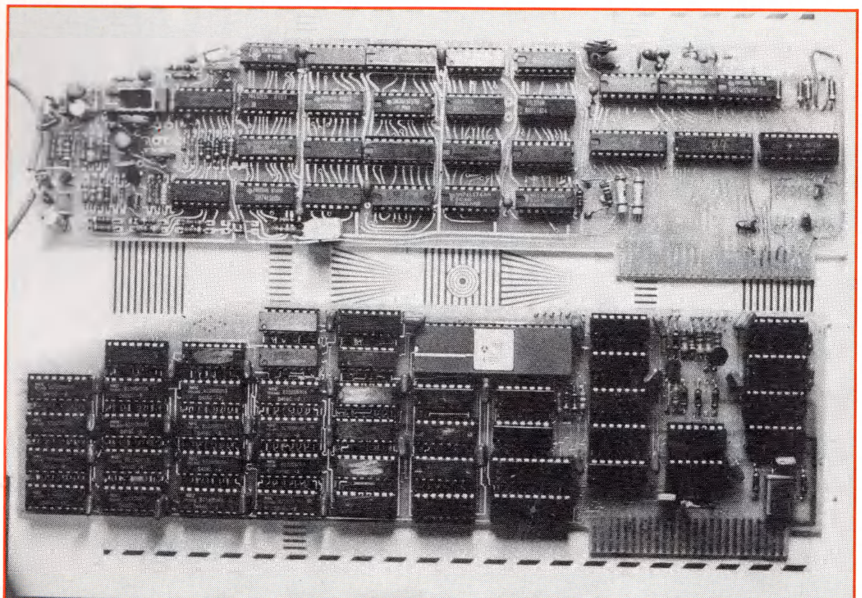
Zutreffendes ankreuzen

suche ☐ Software — suche ☐ Hardware — ☐ Tausch — ☐ Versch.
biete an ☐ Software — biete an ☐ Hardware — ☐ Kontakte — ☐ Chiffre

BARTSCHER ELEKTRONIK

Rüsten Sie Ihren Apple auf mit unseren Video-Interfaces!

(auch für Apple kompatible Computer)



AD 16-3
Echtzeitdigitalisierer

1356,– DM
incl. MWSt.

HGR-512
Graphicinterface

1356,– DM
incl. MWSt.

HGR-512 Graphicinterface

- Eigener 32 k Byte Bildspeicher
- Graphic-Controller EF 9365
- Umschaltbar 512 x 512 Punkte in schwarz-weiß oder 256 x 256 in 16 Graustufen bzw. 16 Farben auf jedem dieser Punkte
- Hardcopymöglichkeit von 512er-Modus
- ASCII-Zeichensatz in Größe und Richtung programmierbar
- Vektorplot mit 1 Millionen Punkte pro Sekunde
- Ausgang für RGB- und BAS-Monitore
- Demosoftware auf Diskette

AD 16-3 Echtzeitdigitalisierer

- Echtzeitdigitalisierer in 40 m sec.
- ca. 65 000 Bildpunkte mit 16 Graustufen
- eingebauter DA-Wandler zur Betrachtung des digitalisierten Bildes
- Digitalisierung von Videorecorder, Kamera oder Fernseher
- Keine Software für Bildlesevorgang nötig

NEUANKÜNDIGUNG:

AD 16-4

Echtzeitdigitizer mit
Sofortdisplay und
eigenem Bildspeicher

Digitalisierung und Display geschieht in 1/50 sec.
Karte enthält eigenen Bildspeicher.
Auflösung: 256 x 256 Bildpunkte.
Fremdsynchronisation der Kamera notwendig.



mit AD16-3 digitalisiert durch HGR-512 dargestellt

Bartscher Elektronik · Westring 84 · 3440 Eschwege · Tel.: 0 56 51 / 17 95

WICOSOFT

Mikrocomputer SOFTWARE

WICO·SOFT

Programm-Zusendungen erwünscht!

Neu:

Demnächst Software für alle Systeme lieferbar!

Der Fluch des Pharao



Adventure-Spiel f. VC-20

(Speichererweiterung mind. 16K)

ein Superspiel m. High-Res.-Graphik

Cassette	**19.50 DM**
Diskette	**25.00 DM**

5 Programme für den VC-20 auf einer Cassette!
Geschicklichkeits- und Denkspiele f. Joystick und Tastenbedienung.

- Autorennen
- Rememby
- Roulette
- Mastermind
- Fadenkreuz

alles zusammen nur

19.50 DM

☐ V-Scheck ist beigefügt

☐ per Nachnahme
(zzgl. Nachnahmegebühr)

WICOSOFT
Nordstr. 22, 3443 Herleshausen

Ich bestelle:

	☐ Cassette	19.50 DM
Der Fluch des Pharao	☐ Diskette	25.00 DM
VC-Programmkassette	☐ VC-20 o.Erw.	
	☐ VC-20 +8K	

Name, Anschrift: ______________________